Investigating the Representation of Backchannels and Fillers in Fine-tuned Language Models

Anonymous ACL submission

Abstract

Backchannels and fillers are important linguistic expressions in dialogue, but often treated as 'noise' to be bypassed in modern transformer-based language models. Our work studies the representation of them in language models using three fine-tuning strategies. The models are trained on three dialogue corpora in English and Japanese, where backchannels and fillers are preserved and annotated, to investigate how fine-tuning can help LMs learn their representations. We first apply clustering analysis to the learnt representation of backchannels and fillers, and have found increased silhouette scores in representations from fine-tuned models, which suggests that fine-tuning enables LMs to distinguish the nuanced semantic variation in different backchannel and filler use. We also use natural language generation (NLG) metrics and qualitative analysis to confirm that the utterances generated by fine-tuned language models resemble human-produced utterances more closely. Our findings suggest the potentials of transforming general LMs into conversational LMs that are more capable of producing human-like languages adequately.

1 Introduction

In our daily life, it is quite ubiquitous to encounter conversation such as below:

Alice: Did you check your inbox?
Bob: *uh-huh*.
Alice: Good.
Bob: *uh*, but why?

The italic words in this dialogue are examples for backchannels (*'uh-huh'*) and fillers (*'uh'*), which play an important role in managing the flow of conversation. The use of backchannels/fillers is seen as an important way for interlocutors to negotiate the common ground (Clark, 1996). Since backchannels/fillers usually do not convey information directly and have only pragmatic functions (Jucker and Smith, 1998) (e.g., expressing affirmation or disagreement to the previous utterance), they are semantically 'bleached' elements (Fuller, 2003) and deemed optional as they lack concrete meanings (Schourup, 1999; Fuller, 2003). As a result, the NLP community often treats them similarly to 'stop words', excluding them during pre-processing as a method to clean the data and improve accuracy (e.g., see Sarica and Luo, 2021). For instance, in dependency parsing for spoken dialogue data, studies such as Charniak and Johnson (2001), Jørgensen (2007) and Dobrovoljc and Martinc (2018) report that excluding backchannels/fillers such as *'uh'* and *'er'* in the Switchboard dataset (Holliman et al., 1992) can significantly improve parsing accuracy.

Although this has been a common practice, previous studies have highlighted that backchannels and fillers can contain rich contextual meaning in dialogue: backchannels such as *'yeah'*, *'okay'*, for example, can serve as feedback towards previous speech (Clark, 1996, pp. 32). Fillers, such as *'uh'*, *'um'*, are used as signals of disfluency (Rose, 2015) and reflect the speaker's cognitive processing when searching for the next word (Clark and Fox Tree, 2002), thus indicating the speaker's cognitive load (e.g., see Table 1 of Berthold and Jameson, 1999 and Rose, 2015). Thus, fillers can be regarded as an important signal to the listener that the speaker needs some time to complete the utterance (and wants to hold the turn; Ball, 1975). Further studies using qualitative approaches show that backchannels/fillers, as a source of feedback, play an important role in the incremental updating of dialogue from a semantic perspective, controlling the flow of information during conversation (Bergey and DeDeo, 2024) and improving the mutual understanding between the interlocutors in order to reach a joint goal (Hough et al., 2015; Howes and Eshghi, 2021).

In order to figure out the potential of transforming language models into conversational language models, which can utilise backchannels/fillers, it is

essential to establish how effectively they can be learnt and represented in language models (LMs). The obstacle for answering the question is that most language data in NLP is text-based (Liesenfeld and Dingemanse, 2022; Dingemanse and Liesenfeld, 2022). Well-annotated conversation data is often insufficient in size and quality. Consequently, there is limited contextual information on backchannels/-fillers for LMs to learn from, and thus LM-based automatic speech recognition (ASR) systems will perform poorly in recognising turn taking/holding, which is largely moderated by backchannels and fillers. Moreover, due to the absence of backchannels and fillers during the pre-training phase, most pre-trained LMs have limited knowledge of these elements. Consequently, the content generated by these LMs tends to be text-like and distant from real dialogue in its form[1], which, as result, fails to *'[put] natural in natural language processing'* (Chrupała, 2023) and lack the ability to act as competent dialogue agents that can, for example, produce (i) appropriate backchannels as feedback to user utterances, and are therefore not considered attentive enough to meet human user expectations (Buschmeier and Kopp, 2018), and (ii) natural fillers in their utterances, which are vital for their role in organising speech and their communicative functions in spoken language understanding (Dinkar et al., 2022). As a result, an LM which ignores backchannels/fillers does not just 'clean' the data, it strips the model of the social cues necessary for fluid interaction, and consequently cannot be used for building a competent conversational agent.

Our work presented here therefore addresses **backchannels/fillers as important linguistic phenomena** that are not well reflected in language models and specifically investigates the issue of learning the representations of backchannels/fillers in language models. A solution that proved effective in tackling dialogue phenomena issue, is fine-tuning (see, e.g., Noble and Maraev, 2021). Although there has been a great deal of work attempting to answer the question of how the representation of linguistic knowledge by a language model, such as BERT (Devlin et al., 2019) and GPT2 (Radford et al., 2019), is altered after fine-tuning, we believe our study makes a meaningful contribution, as an initial study trying to answer how and how well language models can represent backchannels/fillers after fine-tuning. Our general research question (**RQ1**) and three more specific research questions are:

RQ1: Can modern transformer-based language models, such as BERT, GPT-2 and a larger language model such as LLaMA-3 8B, Qwen-3 8B learn representations or improve the representations for backchannels/fillers through fine-tuning?

RQ2: What role does contextual information play when we try to obtain the representations of backchannels/fillers from LMs?

RQ3: Which of the studied LMs can benefit more from fine-tuning?

RQ4: Do different fine-tuning strategies make a difference in learning the representation of backchannels/fillers?

2 Related Work

Backchannels/Fillers as Discourse Markers: Backchannels/fillers are considered to be discourse marker (Jucker and Smith, 1998; Jucker, 1998). They are semantically 'bleached' elements in conversation, such as *oh, yeah, uh, uhm* which can be either fillers or backchannels – depending on whether they are within the dialogue as a sign of disfluency, or stand alone or at the beginning of an utterance, which are then usually taken as feedback to the previous utterance.

As indicated by Fuller (2003), Fox Tree (2010) and Skantze (2021), backchannels/fillers have the following two properties: first of all, they indicate the turn relations of utterances and thus play a role in conversation management, e.g., *uh* and *uhm* in speech, as a signal of disfluency, can indicate turn-holding and pause; secondly, they are 'optional', i.e., deleting them from the utterance won't change its truth conditional meaning. Although fillers and backchannels are considered semantically bleached elements in dialogue, they are considered important for managing the flow of the dialogue and play an important role in 'grounding' processes: reflecting the attentiveness of the listeners during the interaction, confirming listeners' understanding state as well as establishing common ground (Clark, 1996; Buschmeier and Kopp, 2018). Moreover, backchannels/fillers are linguistic universals, as shown in the

[1] In our opinion, this case is also applicable to LLM such as LLaMA-3-8B, as evidenced in our observation shown in the examples in Figure 8a to 8d in the Appendix, during the NLG task, the pre-trained LLM barely generates any backchannels and fillers, which should be considered as an important sign that backchannels/fillers are not well represented in the pre-trained language model

survey by Dingemanse and Liesenfeld (2022), they tend to be the most frequent expressions in spoken language distribution.

Given the rich roles backchannels/fillers play in dialogue, some studies look into the prediction and generation of backchannels/fillers. For example, Skantze (2017) reports that an LSTM-based model can predict the occurrence of backchannels in dialogue. Ruede et al. (2017) show that using word embeddings in a speech model can improve the accuracy of backchannel detection. Amer et al. (2023) proposes a transformer-based pipeline to predict backchannels and further use the predicted backchannels as an index of the agreement among interlocutors. Wang et al. (2022) build three language models to generate fillers in clean speech and evaluate how this helps to improve the naturalness of the generated speech.

Fine-tuning and Representation: Fine-tuning is an important step to adapt a pre-trained model to novel downstream tasks and learn representations that are important for downstream tasks. It has been consistently reported that fine-tuning can improve LMs' representation capabilities at different levels of linguistic representation. Mosbach et al. (2020), for example, use three sentence level classification tasks selected from the GLUE benchmark (Wang et al., 2018) as the fine-tuning tasks on BERT (Devlin et al., 2019), and find that fine-tuning can indeed affect the representation of linguistic knowledge in language models, especially the last hidden layers. It has been approved in many previous studies that fine-tuning is indispensable for an LM to perform well in different downstream tasks (e.g., Noble and Maraev, 2021; Merchant et al., 2020). A classical way to evaluate a fine-tuned LM is to use probing techniques to investigate the meaning representation of the hidden layer weights in the fine-tuned model, which can either be a supervised method (*building a classifier to report accuracy*), or an unsupervised method (*using clustering to report clustering quality before and after fine-tuning*) (see, e.g., Zhou and Srikumar, 2021; Mosbach et al., 2020). However, what is unknown is how language inputs that are less represented in the original language model would be represented in the fine-tuned language model[2]. We will try to answer this in our case study of backchannels/fillers, mainly through the use of clustering after the fine-tuning process.

[2]We observed that backchannels/fillers have high token IDs, which indicates that they are not included in the original vocabulary of the tokeniser, or the tokenizer encountered the backchannels/fillers only in a late phase of pretraining.

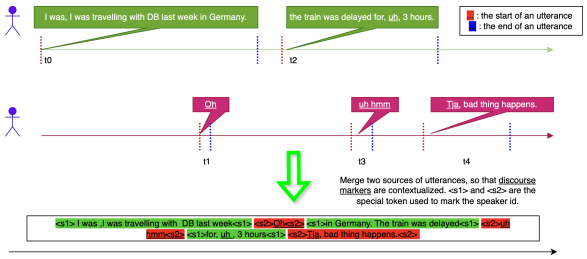


Figure 1: A two-person dialogue example which shows turn-taking and contains various backchannels/fillers. The *uh* from the first speaker is a **filler** while the *oh, uh hmm, and tja* from the second speaker are **backchannels**. We merge the utterances from both speaker (bottom) specifically for the **NTP** and **Masking** fine-tuning tasks: We take the utterances from the two speaker as a whole entity and combined them into one bigger sequence which reflects our consideration that the utterances from one speaker is dependent on the utterance from the other speaker while the speaker information is retained. We add the speaker IDs (e.g., <s1>) as well to let the LM know that the utterances are from two different sources. For the **TTP** fine-tuning task, merging is not required. We use the **TurnGPT** model, which will take the utterances from both speakers in a linear-time order as input and predict which words have higher turn-taking probability.

3 Methodology

3.1 Data Selection

In order to find datasets suitable for this study, we focussed our search specifically on datasets of transcribed spoken dialogue where fillers and backchannels are properly annotated. In order to lower the complexity of the task, the number of interlocutors in the dialogues were be limited to two. We identified the following three datasets which meet our needs: **Switchboard** (Holliman et al., 1992) and **MapTask** (Anderson et al., 1991), both English; and the **BTSJ** 1000 Person Japanese Natural Conversation Corpus (Usami, 2023), which is in Japanese. The two English datasets together have the same size as the Japanese dataset (about 150 000 utterances).

We selected backchannels/fillers based on Todd (2019) and Pilan et al. (2024), which include data of different fillers and backchannels for dialogical interaction in English and Japanese. We report details of the selected backchannels/fillers in Appendix A.

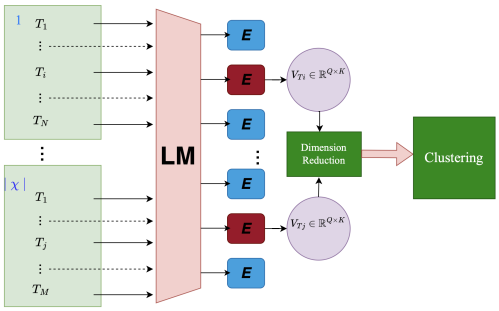


Figure 2: We select the backchannel *'uh huh'* as an example to show how the embedding is obtained from language models. The pipeline consists of the following three steps: (1) Corpus of utterances of varied length, which contain the backchannel *'uh-huh'* in, e.g., T_i, T_j position, in total $|X|$ samples; (2) Text encoding through fine-tuning to get the contextual vector representation of the backchannel *'uh-huh'*, Q is the length of the backchannel tokens, K is then the dimension of the selected hidden state; (3) Reduce the dimension of the contextual vector and build clusters for the contextual vector of the backchannel *'uh-huh'*.

3.2 Task definition

In order to learn the representation of backchannels/fillers in language models, we need to fine-tune existing language models that do not have (or have limited) knowledge of backchannels/fillers using downstream tasks where the models can learn the contextual information of these linguistic items. We use and compare three different fine-tuning tasks: *masking* (**MASK**), *next token prediction* (**NTP**) and *turn taking prediction* (**TTP**).

Masking Masked language modelling encourages models to utilize bidirectional context to build contextualized token representations. Inspired by the pre-training objective of BERT (Devlin et al., 2019)[3], we adapt this method to the conversational setting by selectively masking backchannels/fillers.

Let $\mathbf{X} = (x_1, x_2, \ldots, x_T)$ be a tokenized sequence of utterances drawn from a dialogue, where x_t is the t^{th} token and T denotes the sequence length. We first identify all matching spans $\mathbf{S} = \{s_1, s_2, \ldots, s_N\}$, where each s_n is a continuous token subsequence of backchannels or fillers, and N is the total number of such spans. For each span $s_n \in \mathbf{S}$, define $L = \text{len}(s_n)$, we randomly apply a masking operation $\mathbf{M}$ as follows:

$$M(s_n) = \begin{cases} [\text{MASK}]^L & (P = 0.8) \\ \text{RandomTokens}^L & (P = 0.1) \\ s_n & (P = 0.1) \end{cases}$$

where P denotes the probability of each operation.[4] The resulting corrupted sequence X' is then encoded into contextualized representations $H = (h_1, h_2, \ldots, h_T)$. For each masked span $s_n \in S$, the model predicts the original tokens through (W being the classification layer weights):

$$P(y_{s_{n,k}} \mid X') = \text{softmax}(W h_{s_{n,k}} + b), \quad \forall k \in \{1, \ldots, L\}$$

We use BERT models for English and Japanese from HuggingFace library.

By learning to predict masked backchannels or fillers from contextual discourse, the model can build a better representations. As shown in Figure 1, in order to reflect the notion that backchannels/-fillers are no less different from words with substantial meaning that are conditioned by their previous contexts, in practice we merge the utterances from the two sources of speakers. One of the drawbacks of doing so is that we will miss the information of the speaker of an utterance. As a solution for the fine-tuning task input, we add two special tokens to indicate to the LM the source speakers (<s1> and <s2>, as illustrated in Figure 1). This setting also applied to the fine-tuning strategy, Next Token Prediction.

Next-token Prediction We consider the general task as a language modelling task, i.e., estimating the probability of the next token (e.g., backchannel/filler or other word) given the previous input.

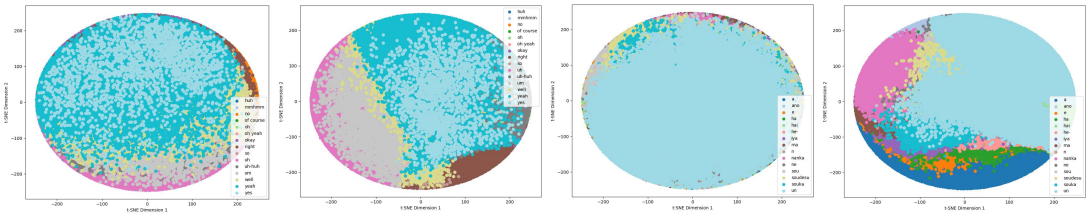
Let $\mathbf{X} = (x_1, x_2, \ldots, x_T)$ represent a sequence of tokens, where x_t is the token at time step t (either a filler/backchannel or a regular word with substantial meaning) and T is the length of the sequence. A pre-trained language model $f(\cdot)$ is fine-tuned to predict the probability of the next token x_{t+1} given all previous tokens $x_1, x_2, \ldots, x_t$. Thus the probability of the next token x_{t+1} given the previous tokens $x_1, x_2, \ldots, x_t$ is expressed as:

$$P(x_{t+1} \mid x_1, x_2, \ldots, x_t) = f(x_1, x_2, \ldots, x_t; \theta),$$

where θ are the model parameters that will be adjusted during the fine-tuning process.

[3] Although BERT is not an autoregressive model like GPT-2 or LLaMA, it is still considered as language model.

[4] For the parameter value P, we use the default value shown in (Devlin et al., 2019)



(a) En_no_ft_one_contxt (b) En_ft_one_contxt (next_token) (c) Jp_no_ft_one_contxt (d) Jp_ft_one_contxt (next_token)

Figure 3: The t-SNE plots of English and Japanese backchannel/filler embeddings from LLaMA-3 model (one context setting, sub-figure a, c). Sub-figures (b,d) show the effect of fine-tuning (**NTP**) in backchannel/filler embeddings.

For this fine-tuning task, we use the Japanese (Zhao and Sawada; Sawada et al., 2024) and English (Radford et al., 2019) GPT-2 models, as well as the multilingual LLaMA-3 8B (AI@Meta, 2024) and Qwen-3 8B (Yang et al., 2025).

Turn-taking Prediction Our third fine-tuning task is turn-taking prediction. The task is chosen due to the close correlation between the use of backchannels/fillers and turn-taking/turn-holding. Specifically, we use the framework TurnGPT (Ekstedt and Skantze, 2020), a language model based on GPT-2 and designed for the prediction of turn-shifts in spoken dialogue. A formal definition of the turn-taking prediction task is as follow:

Let $\mathbf{X} = (x_1, x_2, \ldots, x_T)$ denote a sequence of tokens consisting of the linear ordering of utterances from both interlocutors (e.g., most tokens are from **interlocutor A** while **interlocutor B** produces backchannels/fillers such as *'uh'* or *'um'*), where x_t is the t^{th} token, and T is the sequence length. The fine-tuning task is then to estimate the probability distribution $P(y^* \mid X)$, where $y \in \{1, 2, \ldots, T\}$ indicates the likelihood of a turn-taking event occurring after token x_y. The final predicted turn-taking location is then based on:

$$y^* = \arg\max_y P(y \mid \mathbf{X})$$

Similar to the **NTP** task, training TurnGPT also requires speaker information for each utterance. The details on our data pre-processing are summarised in Appendix B.

3.3 Experiment Set-Up

Our experiment workflow is shown in Figure 2. The chosen LMs are fine-tuned in advance based on the three different fine-tuning tasks[5]. We selected 80% of the conversation data for fine-tuning and the remaining 20% for subsequent generative evaluation. For the evaluation of the representations of backchannels/fillers, we used all the conversation data. We use the backchannel *'uh huh'* as an example to illustrate the workflow. In the initial stage, we pass through all input samples $\mid \chi \mid$ and filter out all utterances that contain *'uh huh'*. We then encode the corresponding utterances through the different hidden layers of a fine-tuned LM, obtaining the representation vector of *'uh huh'* from the final hidden layer. Next, we check the dimensions of the representation matrix. Given that some of the backchannels/fillers can consist of more than one token (as *'uh huh'* in our example), we then apply dimension reduction, simply taking the weighted average value of one dimension of the matrix so that the matrix can be levelled down to a vector representation (embedding). With the embeddings of the backchannels/fillers *'uh huh'*, we examine its representation via clustering. For BERT and GPT-2 models used in the fine-tuning task, the dimension of the hidden layer is 768 for English GPT-2 model and 1024 for the Japanese GPT-2 model. TurnGPT is based on the GPT-2, thus shares the same dimension. A special case is LLaMA-3 8B and Qwen-3 8B as their hidden layer dimension is 4096. For computational reasons, we reduce the dimensionality of the obtained embeddings from all of the models to 100 by using Principal Component Analysis (PCA) (Abdi and Williams, 2010). Technical details of the fine-tuning (e.g., GPU run time, usage of LoRA for parameter optimisation when fine-tuning LLaMA model Hu et al., 2022) can be found in Appendix C.

Moreover, for extracting the embeddings of backchannels/fillers after fine-tuning, we have three different contextual settings: (i) no context information; (ii) one context information; (iii) full context information. In the first setting, when we encounter a backchannel/filler in an utterance, only the utterance containing that backchannel/filler is fed to

[5]Note that in the later stage, when we extract the embeddings of backchannels/fillers for the pre-trained LMs without fine-tuning, the pipeline works in the same way.

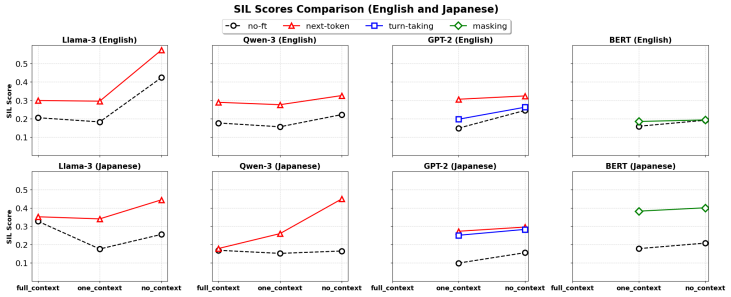


Figure 4: Average silhouette (SIL) scores of backchannels/fillers before and after masking, and two other fine-tuning methods. **NTP** (red line) is applied to GPT-2 based models, Qwen-3 8B and LLaMA-3 8B, while **TTP** (blue line) is only used for GPT-2 based models. For BERT, fine-tuning is performed using the **MASK** (green line) approach. The full context setting is not available for the GPT-2 and BERT based models given the model's input size limitations. Dashed lines are results **without fine-tuning**, solid lines are those obtained after **fine-tuning**.

obtain its embedding. In the second setting, we use the previous and subsequent utterances of a backchannel/filler to build the context before obtaining its embedding. For the third setting, we combine all previous utterances when we encounter an utterance containing a backchannel/filler, using this combination as the input to obtain the embedding of that backchannel/filler. The third setting is only applicable to the LLaMA-3 8B and Qwen-3 8B models since both have no input length limitation.

4 Analysis and Results

Overall observation We chose the top 15 most frequent backchannels/fillers in our Japanese and English data to check how the representation of backchannels/fillers changes after fine-tuning. Among the 15 selected backchannels/fillers in each language, typical examples are: the ones indicating positive or negative feedback as the reference objects, and signals for turn-holding (in English, *'yes', 'uh', 'yeah'* etc.; in Japanese, examples are 'はい'(hai, 'yes' in English), 'うん'(un, 'yeah'), 'ああ'(aa, 'ah') etc.).

We report the preliminary results using t-SNE visualization (Van der Maaten and Hinton, 2008). The embeddings of the 15 selected backchannels/-fillers in each language are the input data. These embeddings are extracted from the last hidden layer of the LMs, before and after fine-tuning, as shown in Figure 2. We have selected t-SNE visualisations from the LLaMA-3 model with a single context setting to demonstrate how fine-tuning (in this instance, NTP) modifies the representation of backchannels/-fillers within the language models (LMs). Figure 3a shows that for English data, when we obtain the embeddings of a selected backchannel/filler from the pre-trained LLaMA-3 model, the distinction between different backchannels/fillers are not clear enough (see Figure 3a). We believe that the large overlap of different data points is due to the fact that the pre-trained LMs have limited knowledge of the backchannel/filler and thus will assign random values to the encountered backchannel/filler. After fine-tuning, clearer distinction among the embeddings of different backchannels/fillers start to appear in the English data (see Figure 3b). The effect of fine-tuning for Japanese data is much clearer, after fine-tuning distinctions between different backchannels/fillers emerge as more colours appear in the t-SNE visualization (Figure 3c and 3d). The t-SNE visualisations from other LMs and different settings can be found in Appendix D.

Analysis Using k-means Clustering As a further analysis, we look at all the corresponding embeddings and apply k-means clustering from scikit-learn individually for backchannels/fillers in the English and Japanese data. The motivation for this analysis is based on the fact that backchannels/fillers, as functional words, often have different pragmatic functions (e.g., indicating agreement, hesitation, etc.; Petukhova and Bunt 2009). A similar idea has been discussed in the annotation work of Figueroa et al. (2022, fig. 4), who reveals that the same backchannel/filler can have several different communicative functions. In terms of our study targets, for example, the Japanese backchannel/filler 'うん'(un, 'yeah') can indicate both 'confirmation' and 'hesitation'. When LMs have limited knowledge of backchannels/fillers and are asked to generate the representation, models will usually give random vector values to them, which will lead to two possible clustering results: either a large k value or

small k value. Therefore, if fine-tuning can improve the representation learning of backchannels/fillers, we should expect that the clustering effect of their embeddings will be more salient after fine-tuning. That is to say, in general we should see an increase in k value, which can indicate that fine-tuned LM's meaning representation can reflect different pragmatic functions of backchannels/fillers if k was small; In contrast, if the k value is initially large, it should be smaller after fine-tuning.

In order to quantitatively analyse the quality of clustering of backchannels/fillers' embeddings before and after fine-tuning, we introduce silhouette scores (Rousseeuw, 1987), which are used to measure the quality of clustering. Given a range of k values, which are used to perform k-means clustering, we calculate the corresponding silhouette coefficient **s(i)** for a data point i, which is a n-dim vector:

$$\mathbf{s(i)} = \frac{b(i) - a(i)}{\max(a(i), b(i))},$$

where $a(i)$ is the average euclidean distance between the embedding of the backchannel/filler i and all other embeddings in the same cluster as i. $b(i)$ is the minimum average euclidean distance from embedding i to all embeddings in any other cluster, $-1 \leq s(i) \leq 1$. The **silhouette score** (**SC**) for a clustering is then measured as the average silhouette coefficient over all n embeddings:

$$\mathbf{SC} = \frac{1}{n} \sum_{i=1}^{n} s(i)$$

The higher the value, the better the clustering quality.

The final result is summarised in Figure 4. In all of the three selected LMs, a general tendency we can observe is that with different fine-tuning strategies, the average silhouette score increases (with the exception of Japanese GPT-2, where the effect of fine-tuning seems not be to obvious). Detailed statistics can be found in in Tables 6 to 12 in Appendix D.

5 Discussion

In Section 4, the t-SNE visualization (Figure 3 reveals that fine-tuning LMs will change the representation of backchannels/fillers in a positive way. The analysis using k-means clustering and silhouette scores further shows that fine-tuning has beneficial effects on the representation of backchannels/fillers. Here, we further discuss the results shown in Section 4 in order to address our research questions.

RQ1: Can modern language models learn/improve representations for backchannels/fillers through fine-tuning? From Figure 4, as well as Tables 6 to 13 in Appendix, we can see that fine-tuning the pre-trained LMs with dialogue data that has contextual information on backchannels/fillers, leads to a general increase of the silhouette scores, which indicates a more salient clustering effect after fine-tuning and serves as a crucial evidence that LMs can learn the representation of backchannels/-fillers with the selected fine-tuning strategies, while supplementary Figure 27 in the Appendix shows that the cluster numbers k become more salient: slight increasing when k was small; decreasing when k was large.

RQ2: Role of contextual information For both fine-tuned and no-fine-tuned LMs, a generally decreasing trend in the average silhouette score is observed with increasing context size for obtaining embeddings (exceptions are LLaMA-3 for Japanese under fine-tuning setting and Qwen-3 without fine-tuning). We interpret this result as follows: When context information is added, LMs tend to treat backchannels/fillers as functional words, whose representations would be compressed into more limited regions of the semantic space. The increased context size encodes more information from the surrounding content words to the backchannels/fillers, which "dilutes" their representations to the average level.

RQ3: Differences among the selected language models In this paper, we targeted four types of language models: BERT and GPT-2 are smaller in size (parameters) and monolingual; LLaMA-3 8B and Qwen-3 8B are much larger and multilingual. It turns, see Figure 4 out that the average silhouette scores of clustering from LLaMA-3 embeddings are best among the four selected LM. Model size, however, does not necessarily correlate with better representation in terms of silhouette, e.g., fine-tuned BERT (which is smallest in size) has comparable results to Qwen-3.

RQ4: Effect of different fine-tuning strategies For the GPT-2-based models, we employed NTP and TTP fine-tuning strategies. The results show that both strategies help the LM learn backchannels/fillers, with no significant differences between the

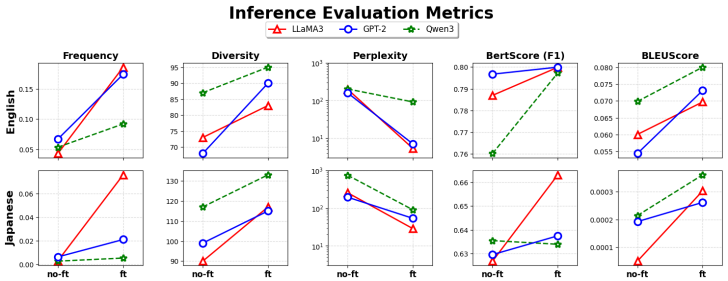


Figure 5: Evaluation metrics of generated backchannels/fillers in the English and Japanese NLG tasks. The models are evaluated before and after fine-tuning with respect to frequency, diversity, frequency-weighted perplexity, BERTScore (F1), and BLEUScore (Metrics definition and details in subsection D.4, Tables 4 and 5 in Appendix)

strategies. However, the embeddings learnt by NTP produce slightly larger silhouette scores (see Figure 4 than those learnt by TTP. Although turn-taking is considered highly relevant to the use of fillers and backchannels, our experiment shows that TTP does not obviously improve the learning of backchannel and filler representations compared to NTP. Another surprising point is that Masking for Japanese BERT model makes a significant improvement on the representation of Japanese backchannels/fillers, which is comparable to the equivalences of LLaMA-3 8B and Qwen-3-8B.

6 Further Analysis via NLG Evaluation

Besides using the silhouette score to evaluate the representation learning of backchannels/fillers in LMs, we also examine generation results to provide evidence of improved representation after fine-tuning. For evaluation, we randomly selected 20% of utterances from both the English and Japanese corpora. The generation task is defined as follows: given two turns of a dialogue in English or Japanese, both pre-trained and fine-tuned LMs are guided to continue writing the dialogue. Based on the generated content, we evaluate backchannels/fillers along five dimensions: (i) frequency of backchannels/fillers, (ii) diversity of backchannel/filler types, (iii) frequency-weighted perplexity of generated backchannels/fillers, and (iv) BERTScore (F1) and (v) BLEUScore against the ground-truth responses. The results are summarized in Figure 5, with detailed results provided in Tables 4 and 5 in Appendix.

After fine-tuning, we observe a general increase in both the frequency and diversity of generated backchannels/fillers, as well as improvements in BERTScore (except for Qwen-3 Japanese) and BLEUScore. At the same time, the frequency-weighted perplexity of backchannels/fillers decreases, offering further evidence that LMs achieve better representations of backchannels/fillers after fine-tuning. We use six illustrative examples in Figure 8 to provide precise human evaluation on the generated dialogues compared to ground truth. As we were concerned that fine-tuning LMs in this study might undermine the LM's general capability of language understanding, we investigate this issue with an additional task (reported in Table 3 in the Appendix), which indicates minor side effect.

7 Conclusion

We investigate the representations of backchannels/fillers in dialogue corpora learned by transformer-based language models, through three different fine-tuning strategies, masking (MASK), next token prediction (NTP) and turn-taking prediction (TTP). The main findings are: Firstly, fine-tuning results in more salient representations of backchannels/fillers as evidenced by the increased clustering performance in semantic space. Secondly, fine-tuned LMs generate utterances that are closer to actual human dialogue, as evidenced by higher backchannel/filler frequency and diversity, lower perplexity on these tokens, and improved similarity to ground-truth conversations (higher BLEU and BERTScore).

Our findings suggest that although backchannels/fillers are typically considered semantically bleached and having only pragmatic functions, their semantic representations are affected by dedicated fine-tuning tasks that incorporate more context information – in a similar way to content words that have concrete meanings. From a broader perspective, this is a case study to investigate LMs' capability of learning under-represented tokens in training data. In a narrow sense, we focus on LMs' capacity of representing backchannels/fillers, which shows the potentials and challenges in developing LMs that can mimic human-like speech styles.

Limitations

Within the scope of this study, we consider the following limitations, which we believe can be further addressed in future work. First of all, at the beginning of the data selection for our experiment, we did consider including language resources such as the German corpora Verbmobil (VM2) (Kay, 1992) and MUNDEX (Türk et al., 2023) in order to give our results a broader linguistic basis. However, in the end we exclude the German corpora due to their comparatively small size and less formatted annotation, which, for now, leaves us with only English and Japanese data.

Secondly, the study would be more thorough with additional tests of large language models, such as Gemini and GPT-4. This was difficult given limited computing resources, where fine-tuning language models like LLaMA-3-8B was our limit. Moreover, a further step will be analysing the representation of backchannels/fillers in different hidden layers, instead of focusing on the last hidden layer of the models. There are a number of papers that examine different layers of models to answer the interpretability question (e.g., Jawahar et al., 2019; Zhao et al., 2024). We leave this as future work.

Thirdly, in this paper, we only consider how fine-tuning tasks affect representation learning of backchannels/fillers. We did not study what kind of effects different fine-tuning techniques can bring to the representation learning of backchannels/fillers. We notice that some studies propose different fine-tuning techniques. For example, surgical fine-tuning (Lee et al., 2023) selects subsets of layers to perform fine-tuning while preserving weights in other layers, which can match or outperform commonly used fine-tuning approaches.

Fourth, there is a big gap between language models and speech models. In speech models, even for the same backchannels/fillers, e.g., 'uh' in English, different representations can be expected based on the differences in voice quality, pitch, and emotional state during speech. How vocal signals of backchannels/fillers are represented in speech models is a future study we will consider.

Finally, we chose to conduct qualitative analysis on our generated dialogues from linguistic perspective to check the use of backchannels and fillers (Details reported in Appendix, Section D.3), which we believe adheres to the generally suggested practice when it comes to NLG evaluation: qualitative text analysis is recommended when the goal is to improve the system (van der Lee et al., 2021). However, we currently do not have a large-scale user study, involving comprehensive human evaluation to further support our claim. We plan recruit native speakers of English and Japanese to evaluate the use of backchannels/fillers in the generated dialogues.

Ethics statement

Given the scope of this study, there do not appear to be any ethical issues. All of the data and models used in this study are open-sourced. We checked the content of the selected dialogue data and made sure that there is no leakage of participants' personal information such as name and id. The writing of the experimental code was supported by ChaptGPT and GitHub Copilot.

References

Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

AI@Meta. 2024. Llama 3 model card.

Ahmed Amer, Chirag Bhuvaneshwara, Gowtham K. Addluri, Mohammed M. Shaik, Vedant Bonde, and Philipp Müller. 2023. Backchannel detection and agreement estimation from video with transformer networks. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The hcrc map task corpus. *Language and speech*, 34(4):351–366.

Peter Ball. 1975. Listeners' responses to filled pauses in relation to floor apportionment. *British Journal of Social & Clinical Psychology*.

Claire Augusta Bergey and Simon DeDeo. 2024. From "um" to "yeah": Producing, predicting, and regulating information flow in human conversation.

André Berthold and Anthony Jameson. 1999. Interpreting symptoms of cognitive load in speech input. In *UM99 User Modeling*, pages 235–244, Vienna. Springer Vienna.

Hendrik Buschmeier and Stefan Kopp. 2018. Communicative listener feedback in human-agent interaction: Artificial speakers need to be attentive and adaptive. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems*, pages 1213–1221.

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Second*

Meeting of the North American Chapter of the Association for Computational Linguistics.

Grzegorz Chrupała. 2023. Putting natural in natural language processing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7820–7827, Toronto, Canada. ACL.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge, UK.

Herbert H. Clark and Jean E. Fox Tree. 2002. Using uh and um in spontaneous speaking. *Cognition*, 84(1):73–111.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. ACL.

Mark Dingemanse and Andreas Liesenfeld. 2022. From text to talk: Harnessing conversational corpora for humane and diversity-aware language technology. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5614–5633, Dublin, Ireland. ACL.

Tanvi Dinkar, Chloé Clavel, and Ioana Vasilescu. 2022. Fillers in spoken language understanding: Computational and psycholinguistic perspectives. In *Traitement Automatique des Langues, Volume 63, Numéro 3 : Etats de l'art en TAL [Review articles in NLP]*, pages 37–62, France. ATALA (Association pour le Traitement Automatique des Langues).

Kaja Dobrovoljc and Matej Martinc. 2018. Er ... well, it matters, right? on the role of data representations in spoken language dependency parsing. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 37–46, Brussels, Belgium. ACL.

Erik Ekstedt and Gabriel Skantze. 2020. TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2981–2990, Online. ACL.

Carol Figueroa, Adaeze Adigwe, Magalie Ochs, and Gabriel Skantze. 2022. Annotation of communicative functions of short feedback tokens in switchboard. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1849–1859, Marseille, France. ELRA.

J.R. Firth. 1968. A synopsis of linguistic theory 1930-1955. In F.R. Palmer, editor, *Selected Papers of J.R. Firth 1952-1959*, pages 1–32. Longman. Reprinted from Studies in Linguistic Analysis, 1957, pp. 1–32.

Jean E Fox Tree. 2010. Discourse markers across speakers and settings. *Language and linguistics compass*, 4(5):269–281.

Janet M. Fuller. 2003. The influence of speaker roles on discourse marker use. *Journal of Pragmatics*, 35(1):23–45.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910, Online and Punta Cana, Dominican Republic. ACL.

E.C. Holliman, J.J. Godfrey, and J. McDaniel. 1992. SWITCHBOARD: telephone speech corpus for research and development . In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 2, pages 517–520, Los Alamitos, CA, USA. IEEE Computer Society.

Julian Hough, Casey Kennington, David Schlangen, and Jonathan Ginzburg. 2015. Incremental semantics for dialogue processing: Requirements, and a comparison of two approaches. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 206–216, London, UK. ACL.

Christine Howes and Arash Eshghi. 2021. Feedback relevance spaces: Interactional constraints on processing contexts in dynamic syntax. *Journal of Logic, Language and Information*, 30:331–362.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. ACL.

Fredrik Jørgensen. 2007. The effects of disfluency detection in parsing spoken language. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 240–244, Tartu, Estonia. University of Tartu, Estonia.

Andreas Jucker. 1998. *Discourse markers: Descriptions and theory*. John Benjamin.

Andreas H. Jucker and Sara W. Smith. 1998. "And people just you know like 'wow'". Discourse markers as negotiating strategies. In Andreas H. Jucker and Yael Ziv, editors, *Discourse Markers: Description and Theory*, pages 171–201. John Benjamins, Amsterdam, The Netherlands.

Masahito Kawamori, Akira Shimazu, and Takeshi Kawabata. 1996. A phonological study on japanese discourse markers. In *Proceedings of the Korean Society for Language and Information Conference*, pages 297–306. Korean Society for Language and Information.

Martin Kay. 1992. *Verbmobil: A Translation System for Face-to-Face Dialog*. University of Chicago Press, Chicago, IL, USA.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. 2023. Surgical fine-tuning improves adaptation to distribution shifts. In *The Eleventh International Conference on Learning Representations*.

Andreas Liesenfeld and Mark Dingemanse. 2022. Building and curating conversational corpora for diversity-aware language science and technology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1178–1192, Marseille, France. European Language Resources Association.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. ACL.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran.

Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. On the Interplay Between Fine-tuning and Sentence-level Probing for Linguistic Knowledge in Pre-trained Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2502–2516, Online. ACL.

Bill Noble and Vladislav Maraev. 2021. Large-scale text pre-training helps with dialogue act recognition, but not without fine-tuning. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 166–172, Groningen, The Netherlands (online). ACL.

Toshiki Onishi, Naoki Azuma, Shunichi Kinoshita, Ryo Ishii, Atsushi Fukayama, Takao Nakamura, and Akihiro Miyata. 2023. Prediction of various backchannel utterances based on multimodal information. In *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*, IVA '23, New York, NY, USA. Association for Computing Machinery.

Volha Petukhova and Harry Bunt. 2009. Towards a multidimensional semantics of discourse markers in spoken dialogue. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 157–168.

Ildiko Pilan, Laurent Prévot, Hendrik Buschmeier, and Pierre Lison. 2024. Conversational feedback in scripted versus spontaneous dialogues: A comparative analysis. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 440–457, Kyoto, Japan. ACL.

Livia Qian and Gabriel Skantze. 2024. Joint learning of context and feedback embeddings in spoken dialogue. In *Interspeech 2024*, pages 2955–2959.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ralph L Rose. 2015. Um and uh as differential delay markers: The role of contextual factors. In *Proceedings of Disfluency in Spontaneous Speech (DiSS). The 7th Workshop on Disfluency in Spontaneous Speech*, pages 73–76.

Peter J. Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20:53–65.

Robin Ruede, Markus Müller, Sebastian Stüker, and Alex Waibel. 2017. Enhancing backchannel prediction using word embeddings. In *Interspeech*, pages 879–883.

Serhad Sarica and Jianxi Luo. 2021. Stopwords in technical language processing. *PLOS ONE*, 16(8):1–13.

Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. 2024. Release of pre-trained models for the Japanese language. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13898–13905. https://arxiv.org/abs/2404.01657.

Lawrence Schourup. 1999. Discourse markers. *Lingua*, 107(3):227–265.

Gabriel Skantze. 2017. Towards a general, continuous model of turn-taking in spoken dialogue using LSTM recurrent neural networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230, Saarbrücken, Germany. ACL.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26:339–373.

James Allen Todd. 2019. "it has the ability to make the other person feel comfortable": L1 japanese speakers' folk descriptions of aizuchi. *Lingua*, 230:102737.

Olcay Türk, Petra Wagner, Hendrik Buschmeier, Angela Grimminger, Yu Wang, and Stefan Lazarov. 2023. Mundex: A multimodal corpus for the study of the understanding of explanations. In *Book of Abstracts of the 1st International Multimodal Communication Symposium*.

Mayumi Usami, editor. 2023. *Building of a Japanese 1000 Person Natural Conversation Corpus for Pragmatic Analyses and Its Multilateral Studies, and NINJAL Institute-Based Projects: Multiple Approaches to Analyzing the Communication of Japanese Language Learners*. NINJAL.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 86:2579–2605.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. ACL.

Siyang Wang, Joakim Gustafson, and Éva Székely. 2022. Evaluating sampling-based filler insertion with spontaneous TTS. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1960–1969, Marseille, France. European Language Resources Association.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Tianyu Zhao and Kei Sawada. rinna/japanese-gpt2-medium.

Zheng Zhao, Yftah Ziser, and Shay B Cohen. 2024. Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15195–15214, Miami, Florida, USA. ACL.

Yichu Zhou and Vivek Srikumar. 2021. DirectProbe: Studying representations without classifiers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5070–5083, Online. ACL.

A Data Statistics

In this section, we include details on the different backchannels/fillers used in this paper, as shown in Tables 1 and 2. The Japanese data is selected on the basis of the guidance and analysis shown in Kawamori et al. (1996) and Onishi et al. (2023). The English data is based on data descriptions in the Switchboard Corpus (Holliman et al., 1992). In the main text, we selected the 15 most frequent backchannels/fillers for our data analysis. In the MapTask (Anderson et al., 1991) and Switchboard datasets, we select the following backchannels/-fillers as our study objects. In the BTSJ 1000 Person Japanese Natural Conversation Corpus (BTSJ) (Usami, 2023) dataset, we use the following backchannels/fillers as our study objects. It is important to note that the backchannels/fillers without comma are used to represent its different variants, for example 'うん'(un, 'yeah') includes 'うん、'(un,) and 'うん。'(un.).

Table 1: The top 15 selected English backchannels/fillers and their frequency in **Switchboard** and **MapTask** (combined), which contain 127672 backchannels/fillers.

Backchannel/Filler	**Occurrence**
uh	24.56%
yeah	17.36%
uh-huh	13.46%
well	7.21%
right	6.70%
oh	5.81%
um	5.49%
okay	3.76%
no	2.63%
yes	1.88%
so	1.08%
oh yeah	0.93%
huh	0.74%
mmhmm	0.71%
of course	0.59%

B Data Pre-processing

As we observed in our data, fillers are usually surrounded by others words with substantial meaning, backchannels on the other hand, usually stand out as an independent utterance. Therefore, our data preprocessing aims to contextualize the backchannels/fillers that are not surrounded by other words

Table 2: The top 15 selected Japanese backchannels/fillers and their English transcription and frequency in the **BTSJ** corpus, which contains 170 898 backchannels/fillers.

Backchannels/Fillers	**Transcription**	**Occurrence**
うん (‘うんうん’, ‘うんうんうん’, ‘ううん’, ‘うーん’, ‘うんー’)	‘un’	23.18%
はい (‘はいー’, ‘はーい’)	‘hai’	17.41%
あ (‘ああ’, ‘あああ’, ‘あー’, ‘あっ’)	‘a’	8.67%
え (‘ええ’, ‘えええ’, ‘えー’, ‘えっ’)	‘e’	6.08%
そう (‘そうそう’, ‘そうそうそう’, ‘そうー’, ‘そーう’)	‘sou’	5.25%
ま (‘まー’, ‘まあ’, ‘まあー’)	‘ma’	4.91%
なんか (‘なんかー’, ‘なんかね’, ‘なんかねー’)	‘nanka’	4.73%
あの (‘あのー’, ‘あのね’)	‘ano’	4.25%
ん (‘んー’)	‘n’	2.67%
そうです (‘そうですね’, ‘そうですねー’, ‘そうですよ’, ‘そうですよね’, ‘そうですよねー’, ‘そーうですね’, ‘そうっすね’)	‘soudesu’	2.40%
は (‘はは’, ‘ははは’, ‘はー’, ‘はあ’, ‘はあー’, ‘はっ’)	‘ha’	2.24%
ね (‘ねー’)	‘ne’	2.19%
いや (‘いやいや’, ‘いやいやいや’, ‘いやー’)	‘iya’	1.77%
へー	‘he-’	1.65%
そうか	‘souk’	1.63%

so that their meaning can be reflected by the other words, i.e., distributional semantics (Firth, 1968, pp. 1-32, Mikolov et al., 2013). The illustrative example which reflects our concept is Figure 1.

We noticed that some of the backchannels/fillers can be subsequences of other words (e.g., *‘um’* is a subsequence of ‘*maxim**um***’), which affects the tokenization of the data when we add the backchannels/fillers in the vocabulary as special tokens. Another case is that some of the backchannels/fillers can be ambiguous, for example, *‘well’* and *‘right’* in English can have substantial meaning as well as serve as feedback with either positive or negative emotion to the previous utterance. The same case holds for the Japanese word ‘ちょっと’ (chotto), which can either mean ‘a little’ or as a backchannels/fillers, indicate, e.g., hesitation. Our solution is to add a special token to the backchannels/fillers so that the tokenizer will not mistake them. This practice is based on our observation in the data that for the backchannels/fillers which are ambiguous, the use of these words as backchannels/fillers usually occur as the first word in the utterance, followed by a comma as a filler (e.g., ‘*well,*’ or co-occur with other backchannels/fillers (e.g., *Okay, right*).

C Implementation Details

We used a total of eight L40 48G GPUs for our experiments, with one GPU assigned to each task. For the BERT model, the runtime for each task is approximately 3 hours for both English and Japanese, which includes fine-tuning and extracting embeddings for the clustering tasks. For fine-tuning GPT-2 Japanese and English, each task of the Japanese experiment takes about 2 hours, which includes fine-tuning and extracting embeddings for the clustering tasks. For English, the tasks take about 1 hours in total. As for the fine-tuning of TurnGPT tasks for Japanese and English, we used the following parameters: batch size 3; weight decay 0.01; dropout rate 0.1; learning rate 0.0001. A total of 15 epochs were used to complete the fine-tuning tasks. After each epoch, a checkpoint (model) was generated, saving the weight parameters gained during training. The model with the minimum loss value was chosen as the final model for estimating the probability of the turn transition potential.

For LLaMA-3 8B and Qwen-3 8B, we used LoRa to accelerate the fine-tuning process. The parameters for LoRA were set as follows. The rank of the low-rank matrices is 16, and a dropout rate of 0.1 is used to the LoRA layers to improve regularization. LoRA is applied to the ‘q_proj’ and ‘v_proj’ layers within the model’s attention mechanism. For LLaMA-3 8B, each task takes about 8 hours in the Japanese data and about 7 hours in the English data. In contrast, the Qwen-3 8B model requires around 15 hours and 10 hours for the same tasks, respectively.

For k-means clustering, we first standardize the obtained embeddings and apply PCA (Principal Component Analysis) to reduce the dimensionality to 100, facilitating subsequent clustering operations. The number of clusters k ranges from 2 to 15, and the optimal k is selected based on the highest silhouette score achieved.

C.1 Other Method for Obtaining Meaning Representation

Parallel to fine-tuning, there are also some other methods to acquire the meaning representations. In terms of learning the representation of backchannels/fillers in LMs, one important method is contrastive learning (Gao et al., 2021). To the best of our knowledge, the only highly relevant work for our study is by Qian and Skantze (2024), who use contrastive learning methods to test how the speech models HuBert (Hsu et al., 2021) and Whisper (Radford et al., 2023), as well as the language model BERT can represent the different functions of feedback signals (namely backchannels). Their results show that the learnt embeddings can carry information about different functions backchannel possesses (see Figure 3 in Qian and Skantze, 2024 for details). In this paper, we focus on fine-tuning only as our method to acquire the representation of backchannels/fillers. We leave contrastive learning aside, since, unlike in Qian and Skantze (2024), the types of backchannels/fillers are much larger and thus induce much higher computational costs. The detailed reason for not using contrastive learning (performed in Qian and Skantze, 2024, while the setting and study objects are different and simpler) to get the representation of backchannels/fillers is mainly due to its computational cost and uncertainty. As an example, in the Japanese data, although we listed the most frequent backchannels/fillers in our paper, we also have the tail examples (those examples with very few occurrence, less than 50 times in the whole dataset) and together we have more than 80 types of backchannels/fillers.

The most important part of contrastive learning is its negative sampling mechanism. Negative sampling works in the way that given a positive example (in our case natural utterance with a backchannel/-filler), we generate several negative samples (so some synthesised utterances where the original backchannels/fillers are replaced with random ones) and use them jointly in training so that positive examples are closer to each other. The reason this was doable and worked well in Qian and Skantze (2024) is that the candidate negative examples are just selected from the backchannels which are classified as feedback, thus limited negative sample candidates and therefore much lower computational cost. Moreover, both positive and negative samples are feedback but with different function types so the experimental results are quite controllable.

D Further Supporting Results

D.1 Representation of Backchannels/fillers From LMs' Different Hidden Layers

In the main text, we evaluate the improvement of representations by applying clustering analysis on the weights extracted from the last hidden layers of the language models. Here we include additional experiments to further investigate the improvement from different layers before and after fine-tuning. To reflect the difference from different hidden layers, for LLama-3 8B, we selected layers 8, 16, 24, 32, for Qwen-3 8B, we selected layers 9, 18, 27, 36, for GPT-2 English and BERT models, we selected layers 4, 8, 12, for GPT-2 Japanese model, we selected layers 6, 12, 18, 24. The results are shown in Figure 6 and Figure 7.

First of all, it turns out that improvement of representation in different selected layers, as quantified by silhouette score, is in general observable in the fine-tuned models in both settings. Second, before fine-tuning silhouette scores are generally higher in the shallow layer than the deeper layers, which indicates that shallow layers have better representations of backchannels/fillers compared to deeper layers. Fine-tuning seems to break this tendency. For example, fine-tuned Llama-3 8b models, under two settings, have improved representation in their deeper hidden layers. Similar patterns can also been seen in Japanese BERT, GPT models and Qwen-3b for English.

D.2 The General Performance of Fine-tuned LMs

Our biggest concern toward the experimental results reported in this study is whether the fine-tuning tasks, used to leverage the usage of backchannels and fillers in LM, will potentially undermine LM's general performance. Therefore, we investigate this issue by checking LM's capability of solving the dialogue act prediction task, which is, in principle, feasible on the **Switchboard** and **MapTask** datasets. A dialogue act is an annotation label on utterance(s) of a dialogue which indicates the communicative functions of the utterance. The communicative functions include *'statement', 'agreement/accept', 'wh-question', 'backchannel/acknowledge'*, etc. (Stolcke et al., 2000). We specifically look before and after fine-tuning whether the accuracy on dialogue act prediction task changes significantly or not. The results are summarised in the Table 3. As can be seen, BERT, GPT-2 and Qwen-3 exhibit a slight de-

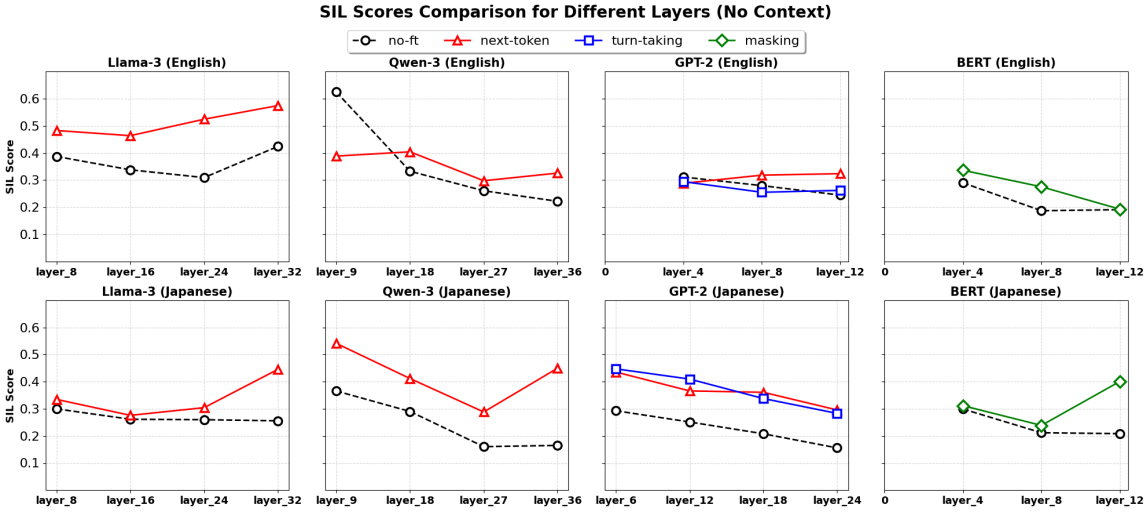


Figure 6: Improvement of the representation of backchannels/fillers from different selected layers under no-context setting.

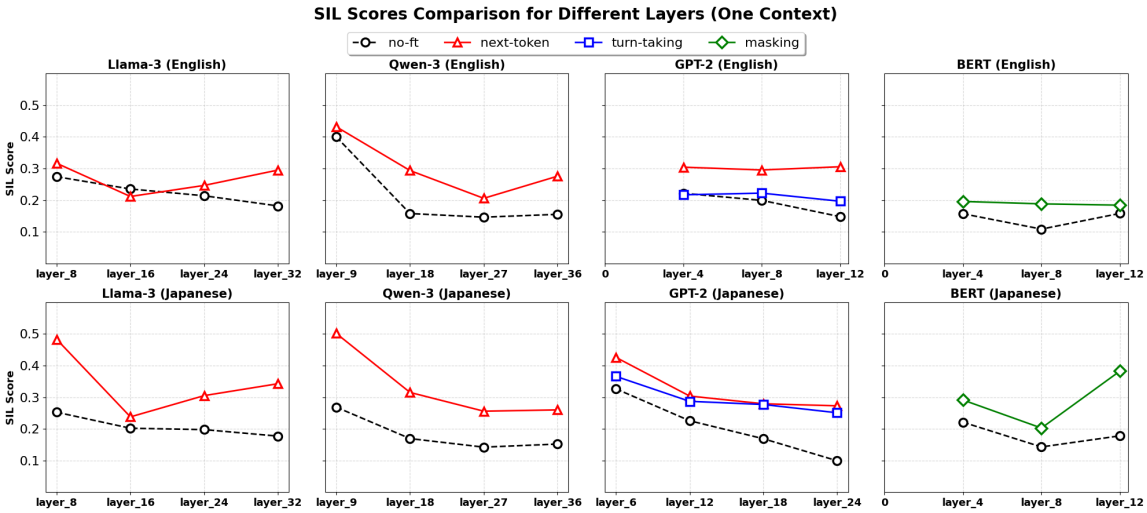


Figure 7: Improvement of the representation of backchannels/fillers from different selected layers under one-context setting.

crease in accuracy after fine-tuning, while LLaMA-3 shows improvements. These results suggest that our fine-tuning strategy does not compromise the language understanding ability of models and may, in some cases, yield modest gains.

Model	non fine-tuned	fine-tuned
BERT	59.0%	58.5%
GPT-2	56.6%	55.7%
LLaMA-3	45.4%	51.2%
Qwen-3	45.3%	42.9%

Table 3: Accuracy on the dialogue act classification task (MapTask dataset) before and after fine-tuning, with linear probing. We use five training epochs for BERT and GPT-2, and three training epochs for LLaMA-3 and Qwen-3.

D.3 Further Qualitative and Quantitative Analysis of the Generation

As further evidence of improved representation capability of backchannels/fillers in fine-tuned LMs, in this subsections, we performs a qualitative analysis on the generation results under the **fine_tuning_no_context** and **fine_tuning_one_context** settings respectively. Both fine-tuned and non fine-tuned LMs are guided to perform an NLG task which requires them to complete the dialogue based on the given context. We selected around 4000 utterances from both English and Japanese corpora. As summarised in Tables 4 and 5, the results show that the fine-tuned LLaMA-3 model increases the usage of backchannels/fillers in the generation task (i.e., the frequency increases). Moreover, different types of backchannels/fillers are used (diversity increases). This is further and crucial evidence showing that the models do learn the representation of backchannels/fillers.

As our small scale human evaluation, here we select six representative dialogue examples (Examples in Figure 8) generated by LLaMA-3 8b model in English as our qualitative analysis on the LLM's capability of generating backchannels and fillers after fine-tuning. Below is the instruction for reading the dialogue examples:

- **/A.../A and /B.../B**: the beginning and the end of the utterance(s) from speaker A, speaker B.
- **///**: the marking of the turn shift
- **<ds>...</ds>**:annotation of the backchannes or fillers.
- **Input:** the incomplete dialogues used for the generation task.
- **Ground truth:** the utterances which are the continuation of the incomplete dialogue. (**Input**)
- **Output (no_ft_one)**: The generated utterances from the off-shelf LLaMa-3
- **Output (ft_one)**:The generated utterances from the fine-tuned LLaMa-3 under the one context setting.

One thing we notice is that, it is only under the **fine_tuning_one_context** setting that LLaMA-3 begins to use backchannels/fillers in its generation, which further indicates the effectiveness of fine-tuning and context information.

The **example 1 and example 4 outputs (ft_one)** show that, fine-tuned LLaMA-3 can generate feedback signals (e.g., **'yeah'**) to acknowledge the previous utterance. It can also use fillers to mimic disfluency or hesitation in its utterance (e.g., **um** in both examples). In **example 2 output (ft_one)**, we can see that, fine-tuned LLaMA-3 can generate a backchannel to show attentiveness to the previous speaker's utterance (the **'uh-huh'** to the utterance by speaker A). In **example 3 output (ft_one)**, we can see a more complex use of fillers: the first filler **'so'** indicates the transition of topic, and again a filler **'um'** indicates disfluency or cognitive load in the production of the speech. The examples shown here all support the idea that we can indeed fine-tune a LM to become a conversational agent which mimics the way human beings produce speech. A notable finding is the emergence of advanced pragmatic competency of backchannels and fillers in the fine-tuned LM. As illustrated in **example 3 and 5 output (ft_one)**, the fine-tuned LLaMA-3 model successfully distinguishes between the structural role of **'so'**, a filler which marks a sequence transition, and the cognitive signalling of the filler **'um'**. Similarly, in **example 5 output (ft_one)**, the **'uh-huh'** is a backchannel which indicates confirmation and signals grounding while **'well'** is a filler which shows transition and framing. Furthermore, as evidenced by **example 6**, the fine-tuned LM does not over-generate backchannels and fillers to complete every dialogue. Instead, it demonstrates a nuanced capacity to omit backchannels and fillers when they are not pragmatically required.

D.4 Supporting Details for NLG Evaluation Result shown in Figure 5

®††Tables 4 and 5 summarize the evaluation results of models generating the next utterance based on a two-turn dialogue context, using the entire 20% evaluation split of the English and Japanese corpora (one utterance per speaker). The generated utterance is constrained to match the length of the ground-truth response. Metrics are defined as follows: (1) **Diversity** counts the number of distinct backchannel/filler types; (2) **Frequency** is the proportion of backchannel/filler tokens in the generated text, normalized by the total number of words for English and by the total number of characters for Japanese; (3) **Perplexity** is the frequency-weighted perplexity computed only on generated backchannel/filler tokens, $\mathrm{PPL} = \exp\big(-\frac{1}{\sum_i f_i}\sum_i f_i \log p(w_i \mid c_i)\big)$, where w_i is a generated backchannel/filler, c_i its context, and f_i its count; (4) **BERTScore (F1)** and (5) **BLEUScore** are computed against the ground-truth continuation. Backchannels/fillers are detected using a curated lexicon with boundary-aware matching. We report results for both the pre-trained (`no_ft_one`) and fine-tuned (`ft_one`) models.

D.5 Extra t-SNE Visualisations

The additional t-SNE figures are shown in Figure 9 to 18. One thing we noticed is that in some of the t-SNE results from LLaMA-3 8B, even in the absence of fine-tuning and contextual information, the distinction between different backchannels/fillers is quite clear. We believe this is the case because the LLaMA models have already learnt some information about backchannels/fillers during their pre-training phase. Another important issue is that, even with fine-tuning, the use of contextual information introduces uncertainty, as some of the visualizations with fine-tuning settings may still have border between different backchannels/fillers.

D.6 Silhouette Scores Statistics and K-Mean Values Before and After PCA Dimensionality Reduction

Tables 6 to 13 respectively show the silhouette scores of each backchannels/fillers in both the original dimensional space and after dimensionality reduction to 100 dimensions using Principal Component Analysis (PCA), along with the average silhouette scores for English and Japanese data. A general trend is that, with fine-tuning for most of the case silhouette score increases. When there is no fine-tuning, adding context size will also have the same effect. However, combining context size with fine-tuning as well as increasing context size will lower the silhouette score. Figure 27 and Table 14 to 21 summarize the change of the k value before and after fine-tuning.

D.7 Confusion Matrices for the Top 15 Backchannels and Fillers in English and Japanese

As our **further prove to the result that fine-tuning helps LMs to learn the representations of backchannels/fillers**, we report the distance among different backchannels/fillers before and after fine-tuning in both English and Japanese using confusion matrices (Figures 19 to 26, darker colour indicates larger distance). The fine tuning strategy is **NTP** and **MASK**. For this additional experiment, we first obtain the centroids of the k-means clustering corresponding to the maximum silhouette score. Each centroid is an embedding value. After getting the centroids for each backchannels/fillers, we calculate the average of the centroids to get an embedding value which can represent the corresponding backchannels/fillers. We calculate the distance between two backchannels/fillers using euclidean distance. In the final visualisation, we can observe that, in GPT-2 based models, some backchannels/fillers are getting closer after fine-tuning (e.g., 'of course' and 'okay' in English) while some are more away from each other (e.g., 'mmhmm' and 'yes', 'no')(Figures 21a and 21b). In the LLaMA-3 8B based models, some of the backchannels/fillers are considered more similar to each other before fine-tuning and the situation has changed after fine-tuning (Figures 19 and 23). And the same is true for Qwen-3 (Figures 20 and 24) and BERT based models (Figures 22 and 26).

Dialogue 1

Input: /A **<ds>yeah</ds>** /A ///
/B so we went in and the financed the car for five and we didn't know that we got kind of taken just a little bit so we kind of upset that we having to you know pay this fifth year on the car because it was just not a wise thing but we learned a good lesson so you know /B ///

Ground truth: /A **<ds>well</ds>** five years is not that **<ds>uh</ds>** five years is not that bad of a problem only because you paying more interest but your payments are lower /A

Output (no_ft_one): /A this all the experience that we got in that situation yeah I just want to say that it was not just the car that we had to deal with but it was /A

Output (ft_one): /A **<ds>yeah</ds>** we were just **<ds>um</ds>** my my parents bought the car that we would drive in high school and they put like two hundred down and and I... /A

Example 1

Dialogue 2

Input: /A but then **<ds>uh</ds>** we also didn't do spend **<ds>uh</ds>** time with them trying to trying to **<ds>uh</ds>** you know just family time /A ///
/B **<ds>uh-huh</ds>**/B ///

Ground truth: /A **<ds>uh</ds>** we had a at a fairly young age we started teaching them to play games and card games and so forth and and **<ds>uh</ds>** **<ds>uh</ds>** but we didn't **<ds>uh</ds>** do it as much or as often as **<ds>uh</ds>** I would have liked to I would like to do that as a as a family and /A

Output (no_ft_one): /A with the family that we have now with the family that we have now. What are you going to do with that kind of logic? I mean, I'd like to solve this problem. But I can't. Can you help me? I have a little bit of time before the family comes over. So, can you give... /A

Output (ft_one): /A I think **<ds>uh</ds>** my father was really into them and we would you know go camping and stuff and **<ds>uh</ds>** you know I think if you do that then it important to spend some time with them you know /A ///
/B **<ds>uh-huh</ds>** /B

Example 2

Dialogue 3

Input: /A so it becomes **<ds>uh</ds>** a general battle of the w- sort of the election of the lesser of two evils I think so /A ///
/B **<ds>yeah</ds>** /B ///

Ground truth: /A I don't know I just **<ds>uh</ds>** I unhappy with it but and well I I sort of semi-fortunate right now I a graduate student so I don't make that much /A

Output (no_ft_one): /A The concept of "lesser of two evils" suggests that when faced with a difficult decision, it's better to choose the option that is less undesirable. In the context of the 202 /A

Output (ft_one): /A it always seems to be that way **<ds>so</ds>** **<ds>um</ds>** what do you think is the other major issue now /A ///
/B **<ds>um</ds>** I I don't know I think.../B

Example 3

Dialogue 4

Input: /A **<ds>yeah</ds>** **<ds>well</ds>** my **<ds>uh</ds>** my **<ds>uh</ds>** probably one of the biggest decisions I think that was very strengthening for our family was rather than have one child make that decision /A ///
/B **<ds>uh-huh</ds>** /B ///

Ground truth: /A than just delegate it I think that they they had a great deal of **<ds>um</ds>** all the brothers and sisters got together and they actually had a conference /A

Output (no_ft_one): /A one year. For one of one or a rather long time I am one I do one I did one I have one I will one one I had one /A
Output (ft_one): /A I think we went to church and I think we sought **<ds>um</ds>** the advice of those that could see it from a different perspective /A

Example 4

Dialogue 5

Input: /A you don't have a stone circle? /A /// /B no /B /// /A **<ds>okay</ds>** do you have a stone circle on the map at all /A /// /B **<ds>um</ds>** i have a ruined monastery /B ///

Ground truth: /A fine **<ds>um</ds>** do you have any obstacle below the telephone kiosk /A

Output (no_ft_one): /A hat's good too we can explore it 1a /A

Output (ft_one): /A you have a ruined monastic **<ds>uh-huh</ds>** **<ds>well</ds>** you're going to go down and then you're going to go right and then you're going to go up over the top of the stone monument and then down again /A

Example 5

Dialogue 6

Input: /A **<ds>eh</ds>** can you dr– actually draw round the /A ///
/B **<ds>well</ds>** **<ds>yeah</ds>** i suppose /B /// /A **<ds>well</ds>** if you could draw round them then/A ///

Ground truth: /B straight down to the giraffes /B
Output (no_ft_one): /B then maybe we can make a plan 1. but then i can't actually get from the other end of the rope bridge to the giraffe /B
Output (ft_one): /B i don't know whether there's a river or not i've got a river which is on the left-hand side /B

Example 6

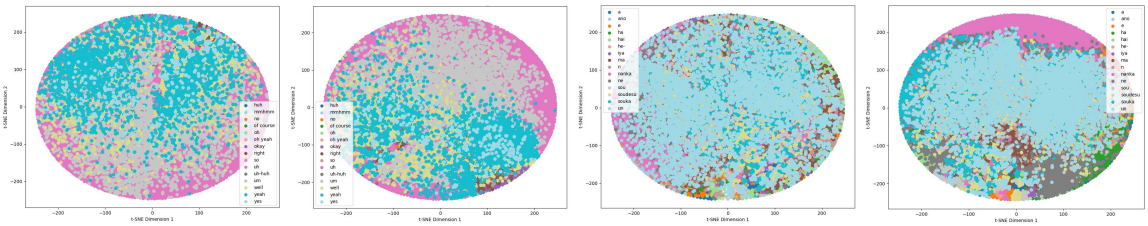
Figure 8: Illustrative examples of generated backchannels/fillers from LLaMA-3.

Metric	LLaMA-3		Qwen-3		GPT-2	
	no_ft_one	ft_one	no_ft_one	ft_one	no_ft_one	ft_one
Diversity ↑	73	83	87	95	68	90
Frequency (%) ↑	4.29%	18.61%	5.33%	9.19%	6.68%	17.43%
Perplexity ⇓	197.67	5.30	202.06	91.32	158.64	6.98
BERTScore (F1 %) ↑	78.69%	79.99%	76.02%	79.73%	79.67%	79.99%
BLEUScore ↑	0.0600	0.0697	0.0698	0.0800	0.0544	0.0731

Table 4: Evaluation metrics for generated backchannels/fillers in **English** NLG task

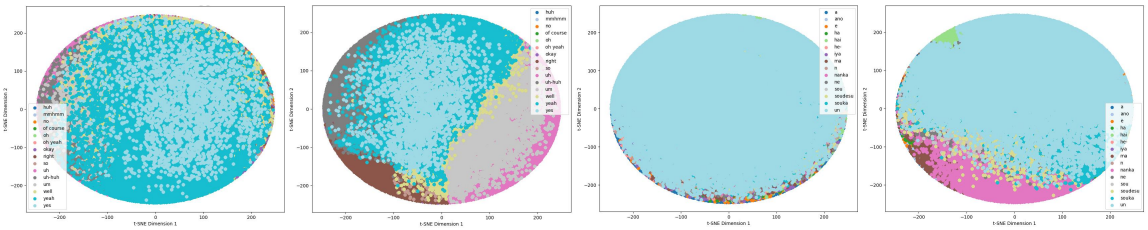
Metric	LLaMA-3		Qwen-3		GPT-2	
	no_ft_one	ft_one	no_ft_one	ft_one	no_ft_one	ft_one
Diversity ↑	90	117	117	133	99	115
Frequency (%) ↑	0.31%	7.57%	0.27%	0.54%	0.63%	2.10%
Perplexity ⇓	255.94	28.51	748.63	90.17	195.36	53.51
BERTScore (F1 %) ↑	62.68%	66.31%	63.55%	63.39%	62.96%	63.75%
BLEUScore ↑	0.00005	0.00030	0.00021	0.00036	0.00019	0.00026

Table 5: Evaluation metrics for generated backchannels/fillers in **Japanese** NLG task



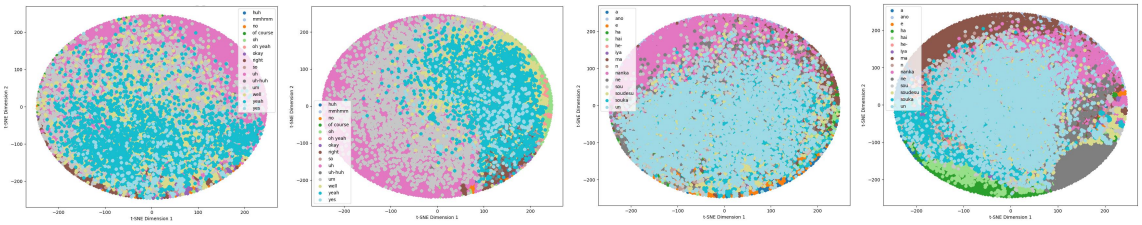
(a) En_no_ft_no_contxt (b) En_ft_no_contxt (next_token) (c) Jp_no_ft_no_contxt (d) Jp_ft_no_contxt (next_token)

Figure 9: The t-SNE plots of the backchannels/fillers embeddings from LLaMA-3 model (NTP) under no context setting.



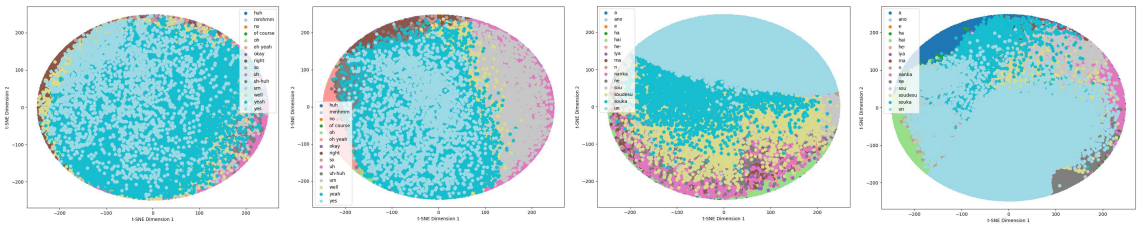
(a) En_no_ft_full_contxt (b) En_ft_full_contxt (next_token) (c) Jp_no_ft_full_contxt (d) Jp_ft_full_contxt (next_token)

Figure 10: The t-SNE plots of the backchannels/fillers embeddings from LLaMA-3 model (NTP) under full context setting.



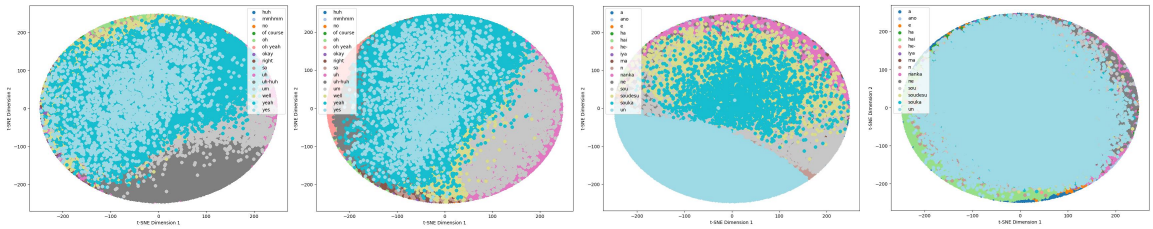
(a) En_no_ft_no_contxt (b) En_ft_no_contxt (next_token) (c) Jp_no_ft_no_contxt (d) Jp_ft_no_contxt (next_token)

Figure 11: The t-SNE plots of the backchannels/fillers embeddings from Qwen-3 model (NTP) under no context setting.



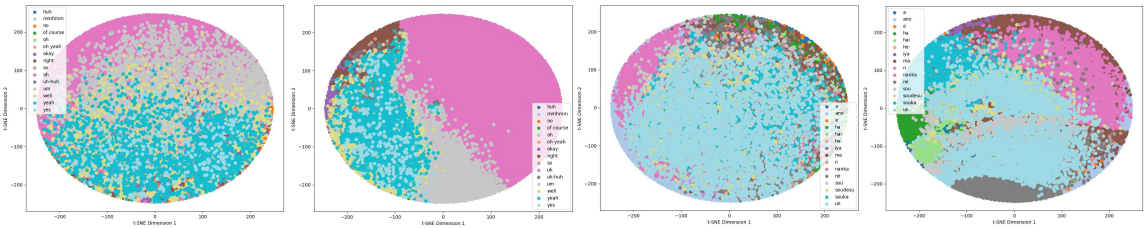
(a) En_no_ft_one_contxt (b) En_ft_one_contxt (next_token) (c) Jp_no_ft_one_contxt (d) Jp_ft_one_contxt (next_token)

Figure 12: The t-SNE plots of the backchannels/fillers embeddings from Qwen-3 model (NTP) under one context setting.



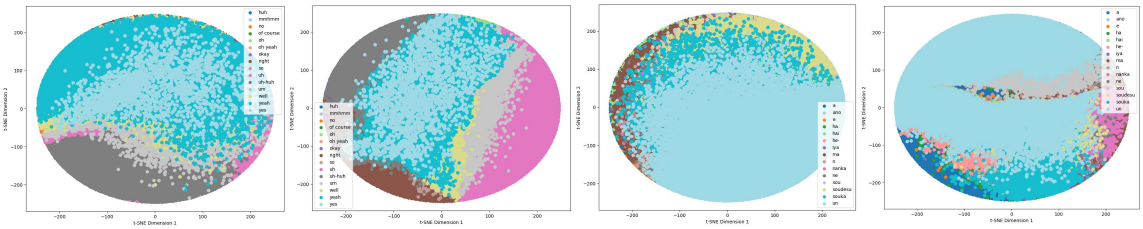
(a) En_no_ft_full_contxt (b) En_ft_full_contxt (next_token) (c) Jp_no_ft_full_contxt (d) Jp_ft_full_contxt (next_token)

Figure 13: The t-SNE plots of the backchannels/fillers embeddings from Qwen-3 model (NTP) under full context setting.



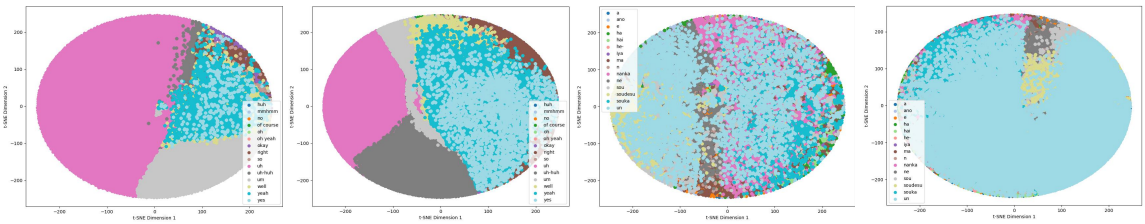
(a) En_no_ft_no_contxt (b) En_ft_no_contxt (next_token) (c) Jp_no_ft_no_contxt (d) Jp_ft_no_contxt (next_token)

Figure 14: The t-SNE plots of the backchannels/fillers embeddings from GPT-2 model (NTP) underthe no context setting.



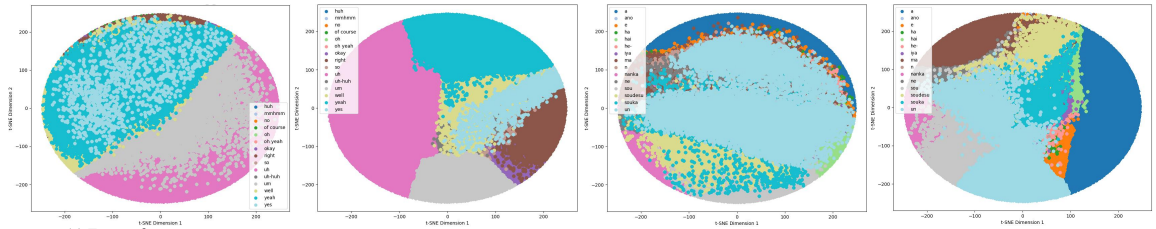
(a) En_no_ft_one_contxt (b) En_ft_one_contxt (next_token) (c) Jp_no_ft_one_contxt (d) Jp_ft_one_contxt (next_token)

Figure 15: The t-SNE plots of the backchannels/fillers embeddings from GPT-2 model (NTP) under one context setting.



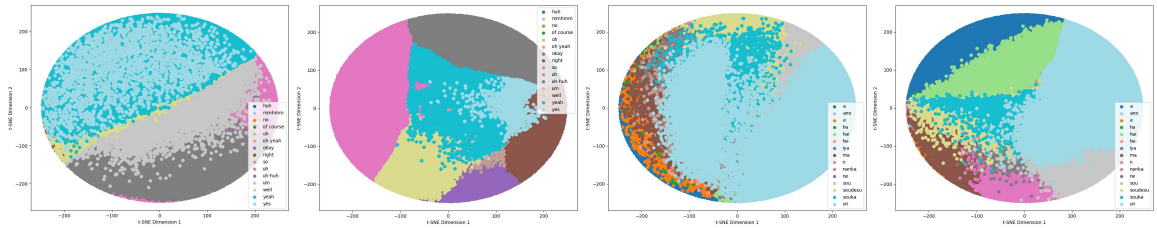
(a) En_ft_no_contxt (turn_taking) (b) En_ft_one_contxt (turn_taking) (c) Jp_ft_no_contxt (turn_taking) (d) Jp_ft_one_contxt (tern_taking)

Figure 16: The t-SNE plots of the backchannels/fillers embeddings from GPT-2 model (TTP) under no and one context setting.



(a) En_no_ft_no_contxt (b) En_ft_no_contxt (masking) (c) Jp_no_ft_no_contxt (d) Jp_ft_no_contxt (masking)

Figure 17: The t-SNE plots of the backchannels/fillers embeddings from BERT model (MASK) under no context setting.



(a) En_no_ft_one_contxt (b) En_ft_one_contxt (masking) (c) Jp_no_ft_one_contxt (d) Jp_ft_one_contxt (masking)

Figure 18: The t-SNE plots of the backchannels/fillers embeddings from BERT model (MASK) under one context setting.

Table 6: Maximum silhouette scores on the k-means clustering of the top 15 selected English backchannels/fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA. **no_ft_no_ctx**: no fine-tuning with no context information; **ft_no_ctx**: fine-tuning with no context information; **no_ft_one_ctx**: no fine-tuning with one context; **ft_one_ctx**: fine-tuning with one context; **no_ft_full_ctx**: no fine-tuning with full context; **ft_full_ctx**: fine-tuning with full context.

Discourse Marker	**no_ft_no_ctx**		**ft_no_ctx (next-token)**		**no_ft_one_ctx**		**ft_one_ctx (next-token)**		**no_ft_full_ctx**		**ft_full_ctx (next-token)**	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
uh	0.302	0.355	0.470	0.500	0.090	0.114	0.057	0.097	0.052	0.233	0.059	0.095
yeah	0.418	0.423	0.565	0.579	0.195	0.205	0.319	0.387	0.084	0.194	0.318	0.392
uh-huh	0.470	0.490	0.606	0.623	0.232	0.191	0.363	0.431	0.147	0.222	0.372	0.437
well	0.403	0.417	0.519	0.479	0.226	0.216	0.186	0.349	0.227	0.275	0.292	0.359
right	0.403	0.433	0.596	0.572	0.211	0.206	0.402	0.496	0.182	0.202	0.410	0.506
oh	0.337	0.364	0.506	0.513	0.205	0.206	0.239	0.279	0.121	0.182	0.226	0.278
um	0.335	0.374	0.482	0.516	0.109	0.127	0.076	0.149	0.054	0.102	0.074	0.144
okay	0.420	0.425	0.583	0.563	0.170	0.203	0.323	0.391	0.170	0.295	0.346	0.427
no	0.424	0.438	0.591	0.580	0.207	0.229	0.262	0.363	0.128	0.167	0.294	0.415
yes	0.397	0.422	0.600	0.598	0.159	0.191	0.267	0.358	0.155	0.280	0.276	0.372
so	0.332	0.339	0.538	0.579	0.163	0.142	0.376	0.477	0.101	0.120	0.386	0.485
oh yeah	0.525	0.359	0.676	0.620	0.183	0.192	0.089	0.141	0.154	0.208	0.078	0.135
huh	0.434	0.461	0.590	0.632	0.186	0.184	0.129	0.161	0.099	0.153	0.107	0.137
mmhmm	0.559	0.505	0.661	0.690	0.232	0.230	0.123	0.247	0.306	0.362	0.103	0.217
of course	0.350	0.376	0.522	0.580	0.081	0.091	0.060	0.102	0.034	0.079	0.048	0.087
Average ↑	**0.407**	**0.424**	**0.567**	**0.575**	**0.177**	**0.182**	**0.218**	**0.295**	**0.134**	**0.205**	**0.226**	**0.299**

Table 7: Maximum silhouette scores value on the k-means clustering of the top 15 selected English backchannels/fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx		no_ft_one_ctx		ft_one_ctx		no_ft_full_ctx		ft_full_ctx	
	ori	100	ori	100	ori	100	ori	100	ori	100	ori	100
uh	0.333	0.172	0.126	0.106	0.332	0.144	0.119	0.155	0.286	0.107	0.153	0.174
yeah	0.396	0.252	0.222	0.231	0.312	0.192	0.339	0.203	0.220	0.146	0.391	0.197
uh-huh	0.376	0.224	0.265	0.286	0.320	0.199	0.393	0.445	0.170	0.135	0.444	0.471
well	0.287	0.143	0.847	0.236	0.243	0.129	0.319	0.237	0.442	0.217	0.382	0.221
right	0.322	0.199	0.837	0.266	0.353	0.185	0.427	0.476	0.224	0.270	0.483	0.514
oh	0.262	0.134	0.202	0.234	0.294	0.135	0.269	0.317	0.215	0.122	0.337	0.301
um	0.316	0.156	0.147	0.132	0.295	0.111	0.178	0.222	0.298	0.130	0.202	0.226
okay	0.349	0.176	0.281	0.345	0.304	0.142	0.335	0.382	0.482	0.260	0.403	0.445
no	0.358	0.204	0.204	0.272	0.395	0.200	0.279	0.340	0.208	0.221	0.356	0.418
yes	0.291	0.207	0.825	0.278	0.313	0.132	0.236	0.354	0.521	0.209	0.351	0.219
so	0.322	0.163	0.815	0.266	0.238	0.136	0.257	0.291	0.253	0.253	0.461	0.490
oh yeah	0.417	0.478	0.909	0.742	0.295	0.154	0.083	0.128	0.198	0.110	0.075	0.128
huh	0.356	0.255	0.725	0.676	0.262	0.151	0.162	0.189	0.329	0.145	0.131	0.152
mmhmm	0.389	0.412	0.614	0.635	0.336	0.189	0.106	0.186	0.640	0.216	0.184	0.201
of course	0.272	0.155	0.312	0.185	0.274	0.130	0.202	0.213	0.185	0.099	0.199	0.180
Average ↑	**0.336**	**0.222**	**0.489**	**0.326**	**0.304**	**0.155**	**0.247**	**0.276**	**0.311**	**0.176**	**0.304**	**0.289**

Table 8: Maximum silhouette scores value on the k-means clustering of the top 15 selected English backchannels/fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (next-token)		ft_no_ctx (turn-taking)		no_ft_one_ctx		ft_one_ctx (next-token)		ft_one_ctx (turn-taking)	
	ori	100	ori	100	ori	100	ori	100	ori	100	ori	100
uh	0.355	0.168	0.443	0.068	0.307	0.110	0.404	0.110	0.475	0.072	0.277	0.109
yeah	0.423	0.206	0.351	0.378	0.356	0.147	0.365	0.160	0.408	0.413	0.277	0.132
uh-huh	0.490	0.282	0.727	0.391	0.271	0.310	0.680	0.172	0.489	0.460	0.299	0.194
well	0.417	0.170	0.414	0.282	0.440	0.311	0.400	0.136	0.332	0.276	0.286	0.123
right	0.433	0.219	0.291	0.388	0.315	0.289	0.406	0.149	0.498	0.487	0.288	0.352
oh	0.364	0.176	0.434	0.205	0.388	0.146	0.363	0.131	0.514	0.330	0.271	0.270
um	0.374	0.150	0.479	0.069	0.349	0.105	0.413	0.105	0.489	0.166	0.314	0.186
okay	0.425	0.213	0.744	0.397	0.282	0.368	0.438	0.161	0.371	0.411	0.237	0.318
no	0.438	0.257	0.746	0.417	0.337	0.309	0.382	0.149	0.334	0.353	0.269	0.219
yes	0.422	0.211	0.753	0.329	0.384	0.229	0.383	0.139	0.360	0.350	0.273	0.188
so	0.339	0.166	0.404	0.395	0.244	0.320	0.486	0.149	0.526	0.457	0.246	0.231
oh yeah	0.359	0.575	0.731	0.559	0.509	0.556	0.374	0.167	0.367	0.262	0.298	0.113
huh	0.461	0.246	0.761	0.337	0.293	0.170	0.366	0.150	0.501	0.194	0.296	0.217
mmhmm	0.505	0.481	0.779	0.549	0.512	0.529	0.442	0.214	0.289	0.268	0.260	0.193
of course	0.376	0.149	0.318	0.094	0.348	0.121	0.422	0.124	0.314	0.096	0.281	0.108
Average ↑	**0.409**	**0.245**	**0.558**	**0.324**	**0.356**	**0.263**	**0.422**	**0.148**	**0.418**	**0.306**	**0.278**	**0.196**

Table 9: Maximum silhouette scores value on the k-means clustering of the top 15 selected English backchannels/fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (masking)		no_ft_one_ctx		ft_one_ctx (masking)	
	ori	100	ori	100	ori	100	ori	100
uh	0.200	0.214	0.086	0.086	0.188	0.167	0.071	0.090
yeah	0.145	0.175	0.147	0.181	0.166	0.156	0.101	0.168
uh-huh	0.148	0.195	0.172	0.197	0.173	0.184	0.115	0.268
well	0.129	0.187	0.167	0.217	0.135	0.159	0.205	0.243
right	0.104	0.132	0.162	0.211	0.140	0.140	0.256	0.305
oh	0.121	0.151	0.171	0.209	0.154	0.151	0.134	0.166
um	0.199	0.221	0.164	0.143	0.195	0.179	0.082	0.111
okay	0.120	0.138	0.150	0.190	0.123	0.124	0.163	0.202
no	0.122	0.169	0.169	0.218	0.128	0.120	0.190	0.240
yes	0.138	0.171	0.140	0.163	0.149	0.140	0.156	0.191
so	0.149	0.201	0.209	0.267	0.141	0.155	0.181	0.280
oh yeah	0.159	0.192	0.157	0.189	0.179	0.164	0.108	0.133
huh	0.214	0.286	0.209	0.246	0.200	0.180	0.098	0.128
mmhmm	0.178	0.239	0.187	0.251	0.137	0.196	0.097	0.128
of course	0.154	0.196	0.093	0.122	0.149	0.158	0.078	0.110
Average ↑	**0.152**	**0.191**	**0.159**	**0.192**	**0.157**	**0.158**	**0.135**	**0.185**

Table 10: Maximum silhouette scores value on the k-means clustering of the top 15 selected Japanese backchannels/-fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (next-token)		no_ft_one_ctx		ft_one_ctx (next-token)		no_ft_full_ctx		ft_full_ctx (next-token)	
	ori	100	ori	100	ori	100	ori	100	ori	100	ori	100
うん (un)	0.265	0.255	0.432	0.442	0.148	0.192	0.246	0.350	0.342	0.348	0.442	0.224
あ (a)	0.264	0.213	0.422	0.427	0.147	0.177	0.237	0.341	0.328	0.300	0.149	0.419
はい (hai)	0.261	0.247	0.440	0.464	0.182	0.184	0.305	0.395	0.331	0.235	0.238	0.321
え (e)	0.290	0.262	0.427	0.430	0.144	0.186	0.251	0.378	0.311	0.233	0.402	0.261
そう (sou)	0.287	0.270	0.429	0.428	0.153	0.184	0.240	0.386	0.324	0.344	0.427	0.291
ま (ma)	0.282	0.242	0.400	0.414	0.114	0.131	0.206	0.316	0.336	0.361	0.139	0.450
なんか (nanka)	0.330	0.309	0.496	0.529	0.111	0.163	0.191	0.286	0.345	0.389	0.472	0.532
あの (ano)	0.262	0.252	0.412	0.427	0.147	0.115	0.162	0.281	0.240	0.247	0.123	0.201
ん (n)	0.291	0.261	0.429	0.428	0.129	0.162	0.307	0.412	0.142	0.339	0.198	0.269
そうです (soudesu)	0.316	0.283	0.410	0.434	0.133	0.228	0.169	0.280	0.349	0.356	0.464	0.175
は (ha)	0.228	0.210	0.352	0.355	0.116	0.164	0.242	0.343	0.265	0.316	0.359	0.315
ね (ne)	0.249	0.246	0.369	0.380	0.115	0.174	0.332	0.436	0.141	0.322	0.311	0.400
いや (iya)	0.283	0.205	0.418	0.474	0.166	0.181	0.246	0.328	0.319	0.358	0.423	0.465
へー (he-)	0.336	0.253	0.533	0.591	0.185	0.221	0.154	0.232	0.405	0.436	0.560	0.605
そうか (souka)	0.319	0.252	0.442	0.455	0.158	0.192	0.271	0.373	0.328	0.369	0.203	0.225
Average ↑	**0.284**	**0.256**	**0.427**	**0.445**	**0.143**	**0.177**	**0.237**	**0.343**	**0.300**	**0.330**	**0.327**	**0.344**

Table 11: Maximum silhouette scores value on the k-means clustering of the top 15 selected Japanese backchannels/-fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

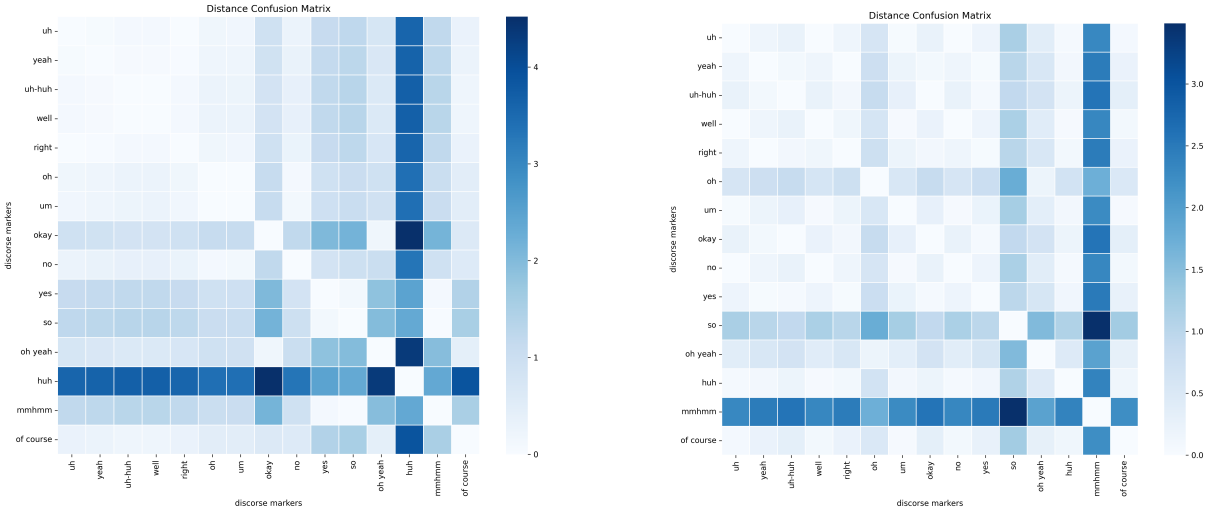
Discourse Marker	no_ft_no_ctx		ft_no_ctx		no_ft_one_ctx		ft_one_ctx		no_ft_full_ctx		ft_full_ctx	
	ori	100	ori	100	ori	100	ori	100	ori	100	ori	100
うん (un)	0.292	0.183	0.666	0.564	0.363	0.186	0.275	0.235	0.189	0.130	0.208	0.154
あ (a)	0.272	0.172	0.157	0.535	0.344	0.151	0.225	0.208	0.291	0.188	0.165	0.168
はい (hai)	0.322	0.242	0.359	0.496	0.356	0.222	0.302	0.240	0.632	0.242	0.168	0.200
え (e)	0.291	0.115	0.740	0.578	0.371	0.134	0.198	0.211	0.213	0.126	0.179	0.154
そう (sou)	0.295	0.191	0.706	0.557	0.353	0.149	0.266	0.289	0.299	0.187	0.175	0.169
ま (ma)	0.295	0.168	0.318	0.290	0.328	0.122	0.335	0.330	0.152	0.149	0.130	0.124
なんか (nanka)	0.243	0.164	0.792	0.153	0.304	0.115	0.155	0.171	0.182	0.172	0.156	0.171
あの (ano)	0.292	0.178	0.770	0.223	0.295	0.131	0.184	0.209	0.201	0.137	0.140	0.161
ん (n)	0.286	0.151	0.739	0.167	0.360	0.155	0.286	0.241	0.252	0.174	0.190	0.144
そうです (soudesu)	0.306	0.122	0.702	0.558	0.343	0.133	0.203	0.267	0.256	0.165	0.150	0.176
は (ha)	0.337	0.227	0.335	0.299	0.365	0.219	0.340	0.313	0.320	0.183	0.204	0.224
ね (ne)	0.298	0.140	0.346	0.405	0.357	0.122	0.450	0.493	0.205	0.114	0.370	0.336
いや (iya)	0.294	0.144	0.742	0.609	0.339	0.419	0.215	0.219	0.198	0.109	0.152	0.162
へー (he-)	0.321	0.151	0.843	0.723	0.406	0.183	0.256	0.223	0.170	0.196	0.182	0.177
そうか (souka)	0.304	0.128	0.732	0.587	0.298	0.106	0.224	0.250	0.178	0.261	0.176	0.154
Average ↑	**0.296**	**0.165**	**0.596**	**0.450**	**0.345**	**0.152**	**0.261**	**0.260**	**0.249**	**0.169**	**0.183**	**0.178**

Table 12: Maximum silhouette scores value on the k-means clustering of the top 15 selected Japanese backchannels/-fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (next-token)		ft_no_ctx (turn-taking)		no_ft_one_ctx		ft_one_ctx (next-token)		ft_one_ctx (turn-taking)	
	ori	100	ori	100	ori	100	ori	100	ori	100	ori	100
うん (un)	0.122	0.153	0.241	0.261	0.302	0.311	0.099	0.103	0.235	0.262	0.316	0.321
あ (a)	0.114	0.119	0.193	0.229	0.256	0.249	0.083	0.081	0.238	0.241	0.292	0.283
はい (hai)	0.186	0.201	0.338	0.390	0.240	0.258	0.107	0.120	0.362	0.388	0.187	0.211
え (e)	0.156	0.150	0.265	0.269	0.293	0.295	0.097	0.087	0.258	0.293	0.319	0.320
そう (sou)	0.146	0.140	0.264	0.305	0.327	0.335	0.122	0.104	0.274	0.312	0.337	0.342
ま (ma)	0.118	0.146	0.204	0.252	0.267	0.278	0.095	0.093	0.194	0.249	0.294	0.303
なんか (nanka)	0.118	0.155	0.103	0.134	0.185	0.195	0.102	0.095	0.119	0.148	0.150	0.174
あの (ano)	0.116	0.160	0.299	0.325	0.268	0.279	0.092	0.093	0.289	0.340	0.290	0.304
ん (n)	0.107	0.133	0.237	0.278	0.275	0.286	0.082	0.079	0.263	0.296	0.299	0.296
そうです (soudesu)	0.136	0.165	0.345	0.369	0.363	0.309	0.119	0.116	0.159	0.178	0.103	0.136
は (ha)	0.135	0.160	0.274	0.313	0.199	0.227	0.121	0.128	0.292	0.330	0.258	0.214
ね (ne)	0.091	0.128	0.284	0.347	0.161	0.208	0.074	0.091	0.288	0.340	0.249	0.250
いや (iya)	0.154	0.163	0.178	0.310	0.226	0.348	0.123	0.100	0.172	0.197	0.164	0.182
へー (he-)	0.187	0.210	0.341	0.375	0.284	0.365	0.104	0.089	0.258	0.284	0.155	0.183
そうか (souka)	0.158	0.152	0.254	0.278	0.249	0.302	0.134	0.110	0.204	0.246	0.208	0.243
Average ↑	**0.136**	**0.156**	**0.255**	**0.296**	**0.260**	**0.283**	**0.104**	**0.099**	**0.240**	**0.273**	**0.241**	**0.251**

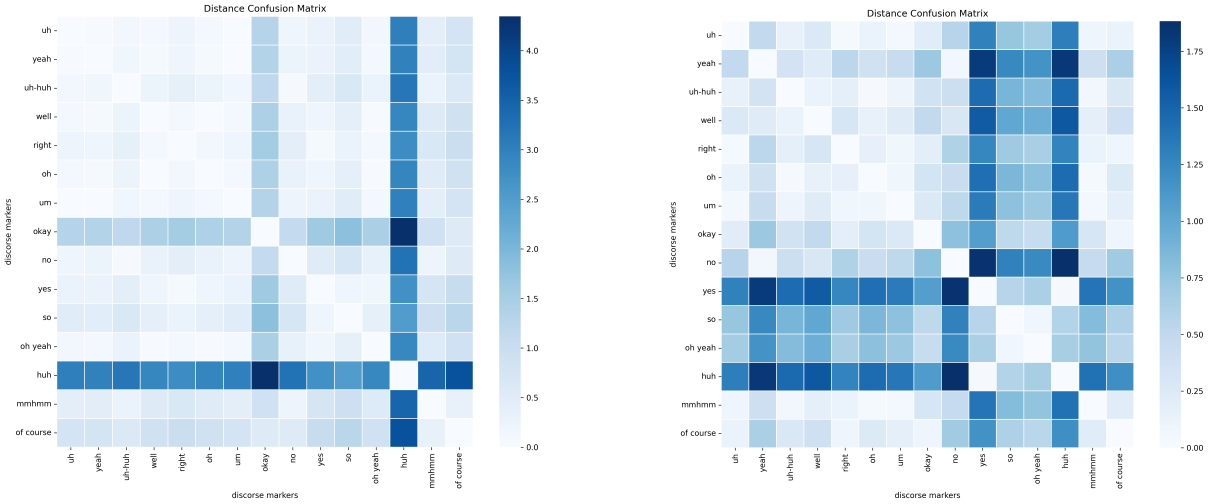
Table 13: Maximum silhouette scores value on the k-means clustering of the top 15 selected Japanese backchannels/-fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	**no_ft_no_ctx**		**ft_no_ctx (masking)**		**no_ft_one_ctx**		**ft_one_ctx (masking)**	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
うん (un)	0.204	0.229	0.377	0.436	0.231	0.211	0.358	0.385
あ (a)	0.134	0.182	0.405	0.443	0.159	0.149	0.370	0.397
はい (hai)	0.241	0.176	0.194	0.376	0.151	0.141	0.159	0.456
え (e)	0.166	0.242	0.302	0.428	0.131	0.244	0.348	0.420
そう (sou)	0.207	0.184	0.362	0.427	0.221	0.140	0.353	0.420
ま (ma)	0.140	0.138	0.385	0.457	0.129	0.070	0.365	0.426
なんか (nanka)	0.094	0.118	0.367	0.103	0.078	0.102	0.357	0.120
あの (ano)	0.113	0.116	0.382	0.299	0.122	0.092	0.373	0.289
ん (n)	0.153	0.107	0.329	0.237	0.191	0.082	0.349	0.263
そうです (soudesu)	0.244	0.136	0.318	0.345	0.133	0.119	0.348	0.159
は (ha)	0.186	0.135	0.393	0.274	0.208	0.121	0.336	0.292
ね (ne)	0.220	0.091	0.327	0.284	0.197	0.074	0.321	0.288
いや (iya)	0.152	0.154	0.390	0.178	0.164	0.123	0.377	0.172
へー (he-)	0.168	0.187	0.139	0.341	0.100	0.104	0.082	0.258
そうか (souka)	0.246	0.158	0.343	0.254	0.290	0.134	0.355	0.204
Average ↑	**0.178**	**0.208**	**0.334**	**0.401**	**0.167**	**0.180**	**0.323**	**0.383**



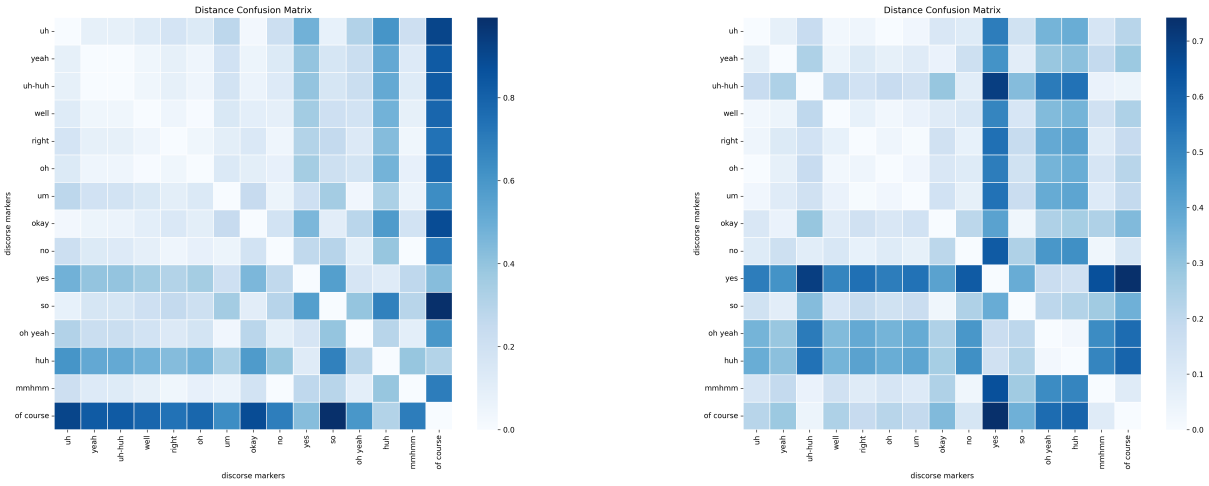
(a) no_ft_one_context (b) ft_one_context (next_token)

Figure 19: Distance matrices of top 15 English backchannels/fillers in LLaMA-3 model before and after fine-tuning.



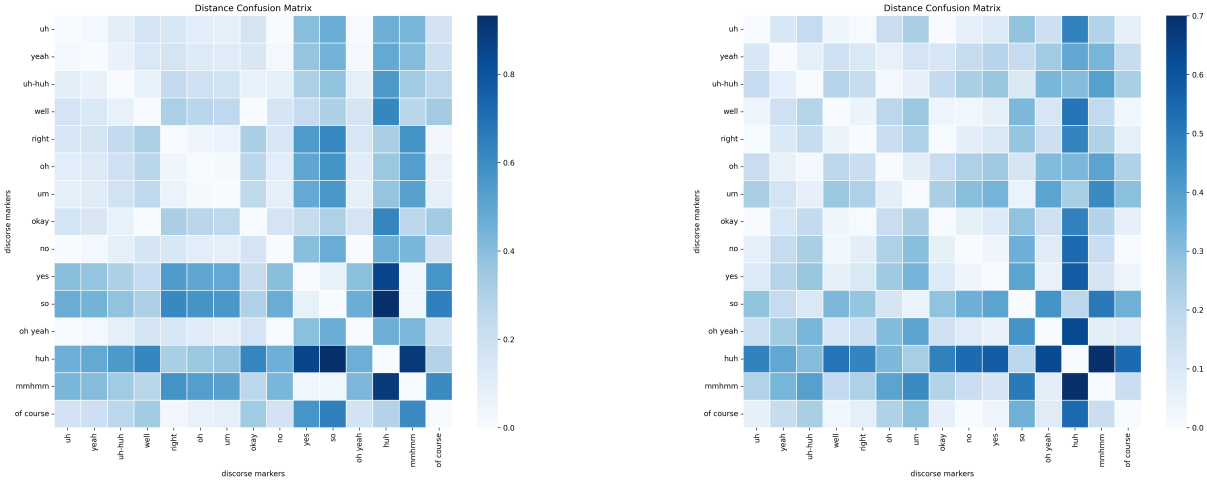
(a) no_ft_one_context (b) ft_one_context (next_token)

Figure 20: Distance matrices of top 15 English backchannels/fillers in Qwen-3 model before and after fine-tuning.



(a) no_ft_one_context (b) ft_one_context (next_token)

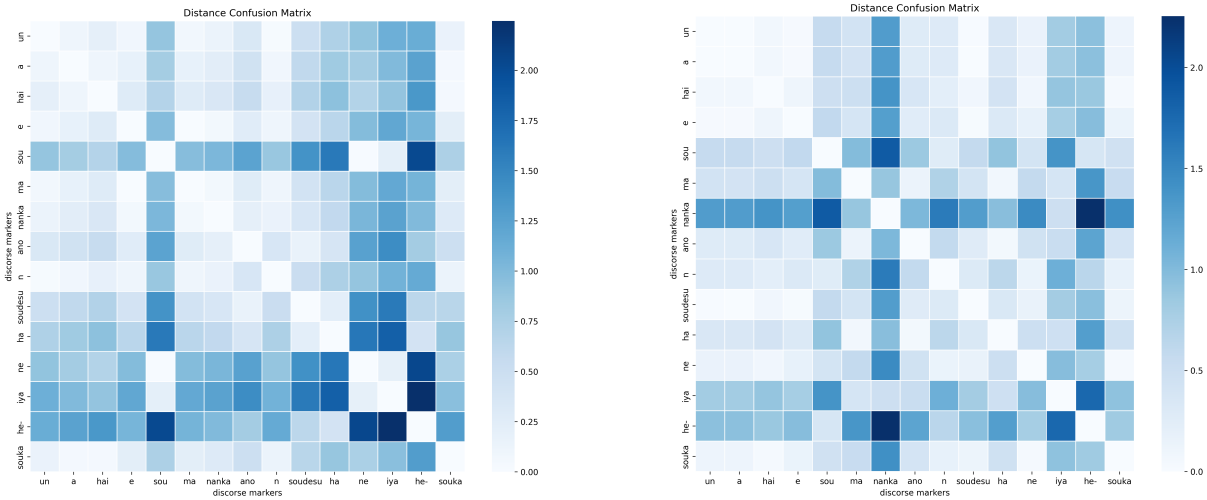
Figure 21: Distance matrices of top 15 English backchannels/fillers in GPT-2 model before and after fine-tuning.



(a) no_ft_one_context

(b) ft_one_context (masking)

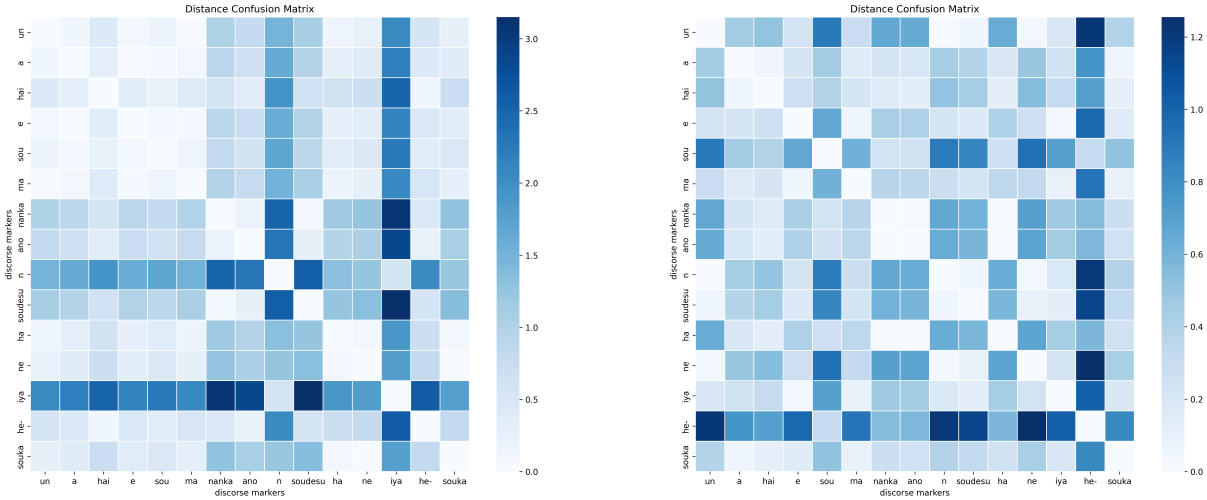
Figure 22: Distance matrices of top 15 English backchannels/fillers in BERT model before and after fine-tuning.



(a) no_ft_one_context

(b) ft_one_context (next_token)

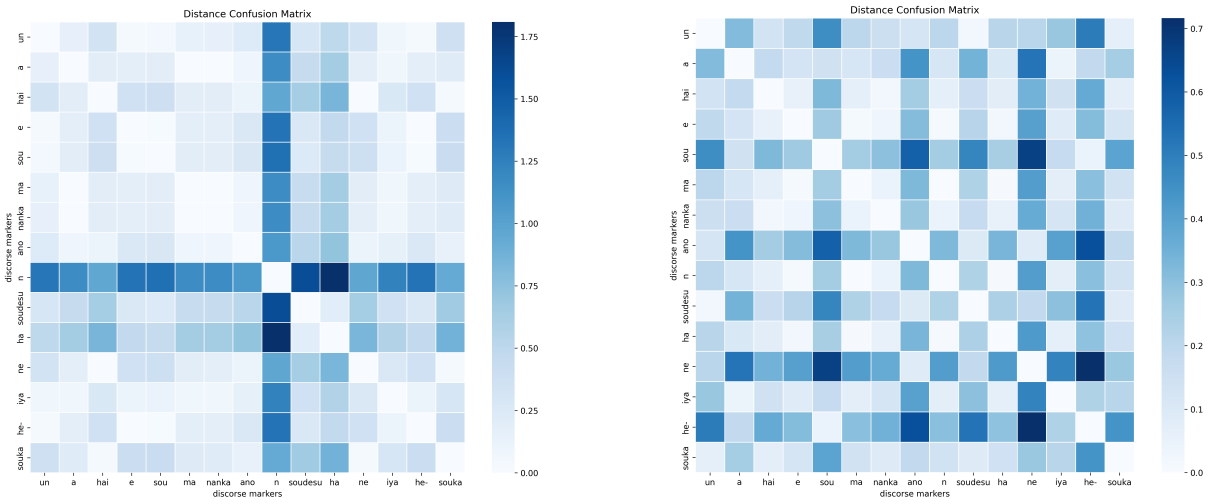
Figure 23: Distance matrices of top 15 Japanese backchannels/fillers in LLaMA-3 model before and after fine-tuning.



(a) no_ft_one_context

(b) ft_one_context (next_token)

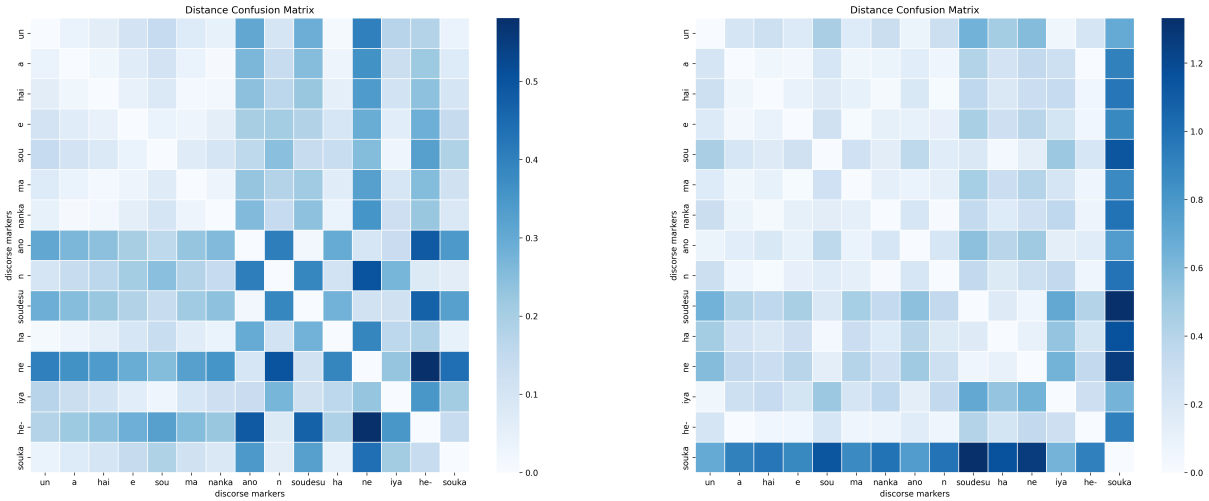
Figure 24: Distance matrices of top 15 Japanese backchannels/fillers in Qwen-3 model before and after fine-tuning.



(a) no_ft_one_context

(b) ft_one_context (next_token)

Figure 25: Distance matrices of top 15 Japanese backchannels/fillers in GPT-2 model before and after fine-tuning.



(a) no_ft_one_context

(b) ft_one_context (masking)

Figure 26: Distance matrices of top 15 Japanese backchannels/fillers in BERT model before and after fine-tuning.

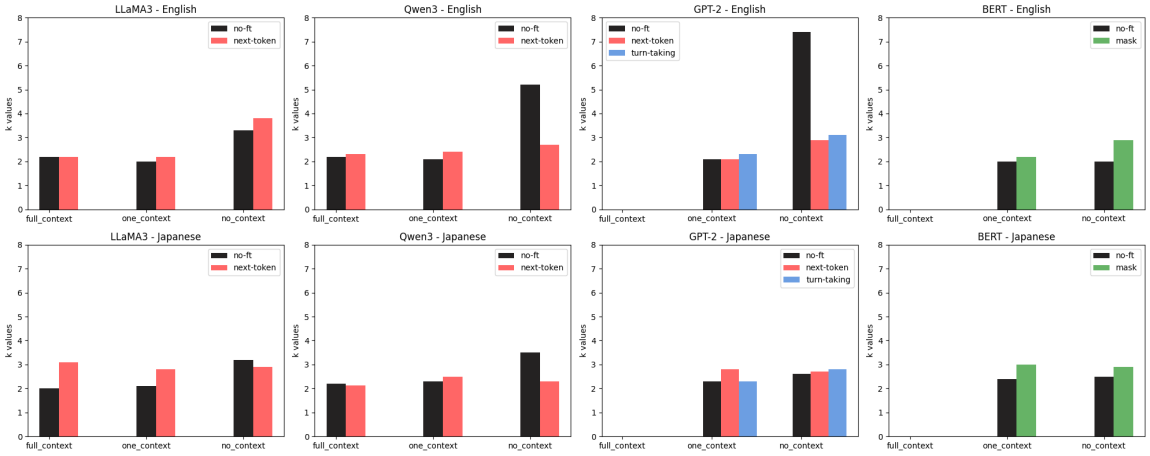


Figure 27: k value change before and after fine-tuning from different LM's results extracted from the clustering analysis. In general, there is a slight increasing of k after fine-tuning among the different fine-tuning strategies. Exceptions are LLaMA-3 for Japanese under the no-context setting; Qwen-3 model under the no-context setting for both language;

Table 14: K values of optimal results on the k-means clustering of the top 15 selected English backchannels/fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (next-token)		no_ft_one_ctx		ft_one_ctx (next-token)		no_ft_full_ctx		ft_full_ctx (next-token)	
	ori	100	ori	100	ori	100	ori	100	ori	100	ori	100
uh	2	2	2	3	2	2	2	2	2	2	2	2
yeah	2	2	3	3	2	2	2	2	3	2	2	2
uh-huh	2	2	3	3	2	2	2	2	2	2	2	2
well	2	2	2	2	2	2	3	2	2	2	2	2
right	2	2	2	2	2	2	2	2	3	2	2	2
oh	2	2	2	3	2	2	2	2	2	2	3	3
um	2	2	3	3	2	2	2	2	3	2	2	2
okay	2	2	2	3	2	2	2	2	3	2	2	2
no	2	2	2	3	2	2	2	2	3	3	2	2
yes	2	2	2	3	2	2	2	2	2	2	2	2
so	2	2	2	3	2	2	2	2	4	4	2	2
oh yeah	3	9	3	11	2	2	4	4	2	2	3	3
huh	2	2	3	3	2	2	2	2	2	2	2	2
mmhmm	2	15	3	9	2	2	3	3	2	2	3	3
of course	2	2	2	3	2	2	3	2	2	2	2	2
Average	**2.07**	**3.33**	**2.40**	**3.80**	**2.00**	**2.00**	**2.33**	**2.20**	**2.47**	**2.20**	**2.07**	**2.20**

Table 15: K values of optimal results on the k-means clustering of the top 15 selected English backchannels/fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx		no_ft_one_ctx		ft_one_ctx		no_ft_full_ctx		ft_full_ctx	
	ori	100	ori	100	ori	100	ori	100	ori	100	ori	100
uh	2	2	2	2	2	2	2	2	2	2	2	2
yeah	2	2	4	3	2	2	2	3	2	3	2	3
uh-huh	2	2	2	3	2	2	2	2	2	2	2	2
well	2	2	2	3	2	2	2	3	2	2	2	3
right	2	5	2	3	2	2	2	2	2	2	2	2
oh	2	4	2	4	2	2	3	3	2	2	2	3
um	2	2	2	2	2	2	2	2	2	2	2	2
okay	2	2	3	3	2	2	2	2	2	3	2	2
no	2	14	5	3	2	2	2	2	2	3	2	2
yes	2	10	2	3	2	3	3	2	2	2	2	3
so	2	2	2	3	2	2	3	3	2	2	2	2
oh yeah	14	15	2	2	2	2	5	3	2	2	4	3
huh	2	2	2	2	2	2	2	2	2	2	2	2
mmhmm	2	12	2	2	2	2	4	3	2	2	2	2
of course	3	2	2	3	2	2	2	2	2	2	2	2
Average	**2.87**	**5.20**	**2.40**	**2.73**	**2.00**	**2.07**	**2.53**	**2.40**	**2.00**	**2.20**	**2.13**	**2.33**

Table 16: K values of optimal results on the k-means clustering of the top 15 selected English backchannels/fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (next-token)		ft_no_ctx (turn-taking)		no_ft_one_ctx		ft_one_ctx (next-token)		ft_one_ctx (turn-taking)	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
uh	2	2	2	2	2	2	2	3	2	2	2	2
yeah	2	2	2	2	2	2	2	2	2	2	2	2
uh-huh	2	4	2	3	2	2	2	2	2	2	2	2
well	2	7	2	2	2	2	2	2	3	3	2	4
right	2	15	2	3	2	2	2	2	2	2	2	2
oh	2	2	2	2	2	4	2	2	2	3	2	2
um	2	2	2	2	2	2	2	2	2	2	2	2
okay	2	15	2	3	2	2	2	2	2	2	3	2
no	2	8	2	3	2	2	2	2	2	2	2	2
yes	3	13	2	2	2	5	2	2	2	2	2	2
so	2	7	2	3	2	5	2	2	2	2	2	4
oh yeah	3	15	2	4	2	3	2	2	2	2	2	2
huh	2	2	2	2	2	4	3	2	2	2	2	2
mmhmm	2	15	2	2	4	7	2	2	2	2	2	2
of course	2	2	2	5	2	2	2	2	2	2	2	2
Average	**2.13**	**7.40**	**2.00**	**2.87**	**2.13**	**3.07**	**2.07**	**2.07**	**2.13**	**2.13**	**2.07**	**2.27**

Table 17: K values of optimal results on the k-means clustering of the top 15 selected English backchannels/fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (masking)		no_ft_one_ctx		ft_one_ctx (masking)	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
uh	2	2	2	2	2	2	2	2
yeah	3	2	4	4	2	2	4	2
uh-huh	2	2	5	3	2	2	4	2
well	2	2	3	4	2	2	2	2
right	2	2	6	6	2	2	2	2
oh	2	2	3	3	2	2	2	2
um	2	2	2	2	2	2	3	3
okay	2	2	2	2	2	2	2	2
no	2	2	2	2	2	2	2	2
yes	2	2	4	3	2	2	2	2
so	3	2	3	3	2	2	4	2
oh yeah	2	2	2	2	2	2	3	3
huh	2	2	2	2	2	2	4	3
mmhmm	2	2	3	3	2	2	2	2
of course	2	2	2	2	2	2	2	2
Average	**2.13**	**2.00**	**3.00**	**2.87**	**2.00**	**2.00**	**2.67**	**2.20**

Table 18: K values of optimal results on the k-means clustering of the top 15 selected Japanese backchannels/fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (next-token)		no_ft_one_ctx		ft_one_ctx (next-token)		no_ft_full_ctx		ft_full_ctx (next-token)	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
うん (un)	3	3	2	3	2	2	2	4	2	2	2	3
あ (a)	3	3	2	3	2	2	4	3	2	2	2	2
はい (hai)	3	5	2	3	2	2	2	2	2	2	3	2
え (e)	3	3	2	3	2	2	2	4	2	2	2	6
そう (sou)	3	3	2	3	2	3	3	5	2	2	2	5
ま (ma)	3	3	2	3	2	2	2	3	2	2	3	2
なんか (nanka)	2	3	3	3	2	2	2	2	2	2	2	2
あの (ano)	2	3	2	2	2	3	2	2	2	2	3	3
ん (n)	3	3	2	2	2	2	2	2	2	2	3	3
そうです (soudesu)	3	3	4	4	2	2	3	2	2	2	2	4
は (ha)	3	4	2	3	2	2	3	3	2	2	2	3
ね (ne)	3	3	2	2	2	2	2	2	3	3	3	3
いや (iya)	2	3	2	3	2	3	2	2	2	2	2	2
へー (he-)	2	3	3	3	2	2	2	2	2	2	2	2
そうか (souka)	3	3	3	3	2	2	4	4	2	2	5	4
Average	**2.73**	**3.20**	**2.33**	**2.87**	**2.00**	**2.13**	**2.53**	**2.80**	**2.13**	**2.00**	**2.53**	**3.07**

Table 19: K values of optimal results on the k-means clustering of the top 15 selected Japanese backchannels/fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx		no_ft_one_ctx		ft_one_ctx		no_ft_full_ctx		ft_full_ctx	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
うん (un)	2	3	2	2	2	2	2	3	2	4	2	2
あ (a)	2	2	2	2	2	2	2	3	2	2	2	2
はい (hai)	2	2	4	2	2	2	2	3	2	2	2	2
え (e)	2	5	2	2	2	2	2	3	3	2	2	2
そう (sou)	2	2	2	2	2	2	2	2	2	2	2	2
ま (ma)	2	2	2	2	2	3	2	2	2	2	2	4
なんか (nanka)	2	2	2	5	2	2	2	4	2	2	2	2
あの (ano)	2	2	2	2	2	2	2	2	2	2	2	2
ん (n)	2	2	2	2	2	2	2	2	2	2	2	2
そうです (soudesu)	2	15	2	2	2	2	2	2	2	2	2	2
は (ha)	2	2	4	3	2	2	2	2	2	2	2	2
ね (ne)	2	2	2	2	2	2	2	2	2	2	2	2
いや (iya)	2	3	2	2	2	2	3	3	3	3	2	2
へー (he-)	2	7	2	2	2	2	2	2	2	2	2	2
そうか (souka)	2	2	2	2	2	5	3	2	2	2	2	2
Average	**2.00**	**3.53**	**2.27**	**2.27**	**2.00**	**2.27**	**2.07**	**2.47**	**2.07**	**2.20**	**2.00**	**2.13**

Table 20: K values of optimal results on the k-means clustering of the top 15 selected Japanese backchannels/fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (next-token)		ft_no_ctx (turn-taking)		no_ft_one_ctx		ft_one_ctx (next-token)		ft_one_ctx (turn-taking)	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
うん (un)	2	4	2	2	2	2	2	4	4	4	2	2
あ (a)	2	2	3	5	2	2	2	2	3	6	2	2
はい (hai)	2	2	2	2	3	2	2	2	2	2	2	2
え (e)	2	2	6	3	2	2	2	2	3	3	2	2
そう (sou)	2	2	2	3	2	2	2	4	2	3	2	2
ま (ma)	2	3	2	2	2	2	2	2	3	2	2	2
なんか (nanka)	2	2	2	2	2	2	2	2	2	2	2	2
あの (ano)	2	2	2	2	2	2	2	2	2	2	2	2
ん (n)	2	3	2	3	2	2	2	3	2	2	2	2
そうです (soudesu)	3	3	2	2	3	3	2	2	5	4	2	3
は (ha)	2	2	2	2	3	3	2	2	2	2	2	3
ね (ne)	3	2	2	2	4	3	2	2	3	3	2	3
いや (iya)	2	5	6	2	6	2	2	2	2	2	2	2
へー (he-)	2	3	2	2	5	2	2	2	2	2	2	3
そうか (souka)	2	2	2	6	5	11	2	2	3	3	3	3
Average	**2.13**	**2.60**	**2.60**	**2.67**	**3.00**	**2.80**	**2.00**	**2.33**	**2.67**	**2.80**	**2.07**	**2.33**

Table 21: K values of optimal results on the k-means clustering of the top 15 selected Japanese backchannels/fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Discourse Marker	no_ft_no_ctx		ft_no_ctx (masking)		no_ft_one_ctx		ft_one_ctx (masking)	
	ori	**100**	**ori**	**100**	**ori**	**100**	**ori**	**100**
うん (un)	3	3	3	3	2	3	2	2
あ (a)	2	4	4	3	3	3	4	3
はい (hai)	2	2	3	3	2	2	5	6
え (e)	4	2	3	4	5	5	4	4
そう (sou)	2	2	2	2	2	2	2	3
ま (ma)	4	4	3	2	4	3	3	3
なんか (nanka)	2	2	2	2	2	2	3	3
あの (ano)	2	3	3	3	2	2	3	2
ん (n)	3	3	2	2	2	2	2	2
そうです (soudesu)	2	2	3	5	2	2	4	4
は (ha)	2	2	5	5	2	2	4	4
ね (ne)	2	2	2	2	2	2	2	2
いや (iya)	2	2	3	3	3	2	3	3
へー (he-)	2	2	2	2	2	2	2	2
そうか (souka)	2	2	3	2	2	2	2	2
Average	**2.40**	**2.47**	**2.87**	**2.87**	**2.47**	**2.40**	**3.00**	**3.00**